

Series:
Text Technology Studies in Pragmatic Psychology
CONTECH 1

**Une théorie et une méthode générale d'analyse textuelle
assistée par ordinateur¹. Le système PROTAN (PROTOCOL ANALYZER)
Version du 2 mars 1995**

(Web page "<http://www.psor.ucl.ac.be/protan/protanae.html>")

Robert Hogenraad
Maître de Recherches au Fonds National de la Recherche
Scientifique

Claude Daubies
Analyste
Université catholique de Louvain

Yves Bestgen
Chercheur Qualifié, Fonds National de la Recherche Scientifique
et Pierre Mahau, Analyste, Université catholique de Louvain

©2003 by R. Hogenraad, C. Daubies, Y. Bestgen, & P. Mahau
Faculté de Psychologie et des Sciences de l'Education
Université catholique de Louvain à Louvain-la-Neuve
10, Place du Cardinal Mercier, B-1348 Louvain-la-Neuve (Belgium)
Tel: ... 32 (0)10 - 47 4411
Fax: ... 32 (0)10 - 47 4834
e-mail: Robert.Hogenraad@psp.ucl.ac.be

¹ La conception du système PROTAN et la rédaction du Manuel ont bénéficié du Crédit aux Chercheurs S 2/5-AO-F62 EX. 1985-1986 1.5.410.86F du Fonds National Belge de la Recherche Scientifique, accordé à Bernard Rimé et Robert Hogenraad.

Ce manuel n'apprend pas l'analyse du texte;
il offre seulement les moyens d'en faire à
ceux qui sont déjà éveillés au texte.

"Ecrire mince", suivant le conseil de Cocteau,
de cette écriture, nue, claire, précise. Car un
manuel peut être économe de ses moyens.

R. H. C. D.

Pour les uns, déchiffrement du silence de la
chose dite. Pour les autres, dont le vocabulaire
est réduit, analyse du contenu.

R.H.

A certains moments de la mise au point du système, nous avons
bénéficié des suggestions de plusieurs collègues.
Nous les en remercions et, avec eux, ceux qui ont manifesté de
l'intérêt pour ce travail.

Table des Matières

	Pages
Préface	4
I. Introduction	5
II. Le système PROTAN comme logiciel	8
III. Description des programmes	10
IV. Les fichiers de données du système PROTAN et leur emploi	23
V. Les programmes du système PROTAN	39
VI. Le programme CDCHECK	41
VII. Le programme CDLISTA	44
VIII. Le programme CDLISTC	46
IX. Le programme CDWJUXT	48
X. Le programme CDWLOOK	54
XI. Le programme CFCHECK	77
XII. Le programme CFLISTA	79
XIII. Le programme CFWKWIC	81
XIV. Le programme CRCHECK	87
XV. Le programme CRLISTA	91
XVI. Le programme CRWSTRIP	93
XVII. Le programme CSCHECK	96
XVIII. Le programme CSCUT	99
XIX. Le programme CSEDIT	109
XX. Le programme CSJOIN	114
XXI. Le programme CSSORT	117
XXII. Le programme CWADD	121
XXIII. Le programme CWEDIT	126
XXIV. Le programme CWFLOW	130
XXV. Le programme CWKWIC	136
XXVI. Le programme CWKWOC	141
XXVII. Le programme CWNEW	146
XXVIII. Le programme CWPAT	151
XXIX. Le programme CWREFER	156
XXX. Le programme CWSELECT	158
XXXI. Le programme CWTALLY	171
XXXII. Le programme CWWCOL	173
XXXIII. Le programme CWWORD	179
XXXIV. Annexes.	
Les dictionnaires de formes nominales et de catégories	188
Les jeux de caractères	194
Le schéma de PROTAN	195
Condensé de la syntaxe des fichiers de données SOURCE .	197
Condensé de la syntaxe des fichiers de données INROOT .	198
Condensé de la syntaxe des fichiers de données INDICT pour dictionnaires de catégories	199
Condensé de la syntaxe des fichiers de données INDICT pour dictionnaires de normes	200
Condensé de la syntaxe des fichiers de données INFIND .	201
XXXV. Abrégés	202
XXXVI. Documentation technique	245
XXXVII. Références	256

Préface

PROTAN (pour Protocol Analyzer) est un système d'analyse du contenu assistée par ordinateur. Ce manuel décrit ce système et donne l'information dont il faut disposer pour pouvoir s'en servir. En coordonnant les modes d'analyse catégorielle et non-catégorielle, ce système réalise d'une certaine façon une théorie de l'analyse du contenu, au sens qu'il la rend opératoire si on veut. Mais il ne décrit pas la théorie de l'analyse du contenu, ni d'ailleurs la ou les théories du discours sur lesquelles repose l'analyse.

Il n'empêche, le lecteur qui cherche une "théorie en action" découvrira dans ce manuel une procédure d'analyse articulée, rapide, et efficace. A ce lecteur s'adressent, eh bien ... tous les chapitres du manuel.

Dans un premier Chapitre, ce lecteur découvrira la place du système PROTAN dans l'ensemble de la méthodologie et de la problématique de l'analyse du contenu. Ce sont, si on veut, les articulations externes du système, auxquelles répondent, au Chapitre II, ses articulations internes, c'est-à-dire la dynamique du système et de l'analyse, c'est-à-dire encore la façon dont les différents programmes se répondent les uns aux autres. Le Chapitre III, lui, fera alors découvrir des aspects déjà plus spécifiques du système, et notamment un aperçu de l'efficacité de chaque programme et de la manière d'activer chacun d'eux.

Les trois premiers Chapitres donnent une idée du fonctionnement général du système. Les deux Chapitres suivants conduisent le lecteur à être capable de créer lui-même tous les fichiers de données, fichiers de programme, et fichiers-système et d'utiliser le système dans ses différentes étapes. Au point que, ayant maîtrisé les Chapitres IV et V, et pris connaissance du Chapitre VI, le lecteur pourra ultérieurement se servir des programmes en se référant seulement aux condensés des programmes qui forment l'essentiel des Annexes (Chapitre XXXIII).

Enfin, si une Grammaire Universelle a été un leurre pour beaucoup, --l'est d'ailleurs encore--, un ordinateur et un système informatique universels ne le sont plus pour personne, s'ils l'ont jamais été. Un système informatique, comme la logique et les institutions, inclut et exclut. On doit donc s'attendre à ce que le système PROTAN aie des limites.

En tout cas, le présent Manuel donne toutes les informations nécessaires pour utiliser PROTAN dans les environnements VM/Batch, DOS, et UNIX. Le dernier Chapitre offre à l'analyste de systèmes la documentation technique nécessaire pour utiliser le logiciel PROTAN ailleurs que dans ces environnements.

Pour conclure. Il vient un moment dans l'analyse pour reconstruire le texte désarticulé et lui redonner un sens inaperçu au premier abord. Et on découvrira alors que "ce qui n'est pas ineffable n'a aucune importance" (Paul Valéry, "Mon Faust", Lust, III, 3) et que l'analyse du contenu est un peu de cet art de faire dire au texte ce qu'il ne veut et surtout ne peut pas dire, de lui faire dire une partie de son silence.

R. H.

1. Introduction

PROTAN (pour Protocol Analyzer) est un système d'analyse du contenu assistée par ordinateur. Ce manuel décrit ce système et les divers éléments dont il faut disposer pour pouvoir s'en servir (dictionnaires de mots et de catégories, dictionnaires de formes nominales et de radicaux, messages, présentation du texte, segmentation du texte en interviews, unités, et locuteurs).

L'acronyme PROTAN a une histoire. Nous avons d'abord pensé à NAG, pour Nouvel Analyseur Général, puisque notre système repose pour une part sur le schéma de l'ancien GENERAL ANALYZER de Donald P. Spence (1969) et pour une autre part sur le système WORDS de Howard P. Iker (1974). Mais NAG était déjà le nom d'un logiciel mathématique. C'est en parcourant l'ouvrage d'Ericsson et Simon (1984), "Protocol Analysis: Verbal Reports as Data", qu'est venue l'idée de l'acronyme PROTAN. Si la méthodologie de l'analyse des protocoles de données verbales d'Ericsson et Simon est sans doute éloignée de celle de l'analyse du contenu, il demeure que la théorie des données verbales de ces auteurs a des retombées immédiates pour les opérations d'analyse du contenu et, en aval, pour l'interprétation de ses résultats.

On pourrait s'interroger sur les raisons d'être du système PROTAN². Beaucoup d'autres systèmes existent déjà. La réponse à cette question nous introduira d'emblée au coeur de toutes les questions liées à la nature de l'analyse du contenu, en particulier assistée par ordinateur. Cette réponse a de multiples facettes. Il faut y mettre de l'ordre.

a. Analyse catégorielle et analyse non-catégorielle.

Quelle que soit son origine, un texte est, comme ce dont est fait l'histoire, un événement unique dont on ne pourra jamais reproduire les conditions de création. On doit donc décrire, on ne peut que décrire (Deese, 1985). Cette limitation ne doit pas empêcher de faire de bonnes descriptions, comme il en existe en sciences sociales ou en anthropologie.

Mais certains peuvent ressentir péniblement la limitation inhérente à toute description, à savoir la portée limitée de ses généralisations. On peut alors concevoir une autre forme de description, non plus dans les termes du texte, mais dans les termes, c'est-à-dire alors les catégories, de l'analyste.

On a ainsi abordé les deux grandes systématisations de l'analyse du contenu, l'analyse non-catégorielle, représentée essentiellement par le système WORDS, de Howard P. Iker, et l'analyse catégorielle, dont les modèles sont nombreux et fort semblables les uns aux autres. On trouvera un développement théorique plus détaillé sur la systématisation de l'analyse du contenu dans un travail récent de Hogenraad (1987) sur la notion de construction en analyse du contenu.

b. Le système WORDS, de Howard P. Iker.

² On pourrait s'interroger aussi sur les motifs qui nous ont poussé à développer une machinerie sophistiquée pour déchiffrer le silence des choses dites, comme l'a écrit Michel Foucault (1969). L'intérêt de la science n'explique pas tout, à moins qu'il faille plutôt expliquer l'intérêt de la science.

Ce système, d'ailleurs le seul représentant de cette orientation de l'analyse du contenu, consiste en une description du texte en termes des relations de contiguïté que les mots, ou des mots, présents dans le texte, ont entre eux. Mais si on considère que le texte est de nature non répétable, est un événement unique donc, il peut se poser de sérieuses difficultés d'interprétation. En effet, au bout du compte, l'analyste doit toujours reconstruire le sens du texte à partir des corrélations et des facteurs qu'il obtient. Ce travail de reconstruction ne va pas sans poser des difficultés semblables à celles que présente l'interprétation d'associations libres dans des protocoles expérimentaux ou cliniques (Spence, 1982).

c. Les systèmes catégoriels.

Les modèles susceptibles d'illustrer l'analyse du contenu catégorielle sont plus nombreux, en partie peut-être parce qu'ils se prêtent à des interprétations moins ambiguës. On y trouve notamment le GENERAL INQUIRER, de Stone, Dunphy, Smith, et Ogilvie, le GENERAL ANALYZER, de Donald P. Spence (en fait, une version simplifiée du premier, tout comme le système COUNT de Colin Martindale), et le système TEXTPACK V, de ZUMA (Zentrum für Umfragen, Methoden und Analysen). D'autres systèmes existent encore bien sûr dans cette orientation.

Avec le système WORDS, on était dans un modèle idiographique (Allport, 1942) de la science; avec ces systèmes-ci, on est dans un modèle nomothétique. On retrouve donc en analyse du contenu les deux modèles de base du fonctionnement de la psychologie, avec les avantages et les inconvénients qui leur sont propres.

La distinction entre ces deux modèles, exprimée dans les termes de la philosophie des sciences, rejoint celle que Vygotsky (1934/1962) établit, en psychologie du langage, entre sens et signification (la traduction anglaise de 1962 parle de "sense of a word" et de "meaning of a word").

"The sense of a word (...) is the sum of all the psychological events aroused in our consciousness by the word. It is a dynamic, complex, fluid whole, which has several zones of unequal stability. Meaning is only one of the zones of sense, the most stable and precise zone. A word acquires its sense from the context in which it appears; in different contexts, it changes its sense" (Vygotsky, 1934/1962, p. 146).

Si on veut, la distinction, en analyse du contenu, entre analyse catégorielle et non-catégorielle, est un cas particulier de la distinction qu'on fait entre sens général et sens particulier donné par le contexte. La signification d'un mot renverrait à un concept général tandis que son sens renverrait aux contextes où il apparaît. Le sens des mots serait donc appréhendé par un système comme le WORDS (analyse des corrélations entre les mots) tandis que leur signification, la partie stable et publique du sens, le serait par des systèmes catégoriels comme TEXTPACK V, le GENERAL ANALYZER, ou le GENERAL INQUIRER.

Dans les modèles nomothétiques, on trouve toujours, en un point de l'analyse, une comparaison des mots du texte aux mots composant chacune des catégories. Rudimentaire, cette procédure d'analyse l'est sans l'ombre d'un doute, mais elle satisfait

l'analyste dans beaucoup de cas. Cette procédure donne en tout cas des informations, (des descriptions?), utiles là où les catégories utilisées sont appropriées à la nature du texte. Construire et recourir à un système approprié de catégories, devient donc une étape cruciale dans ce type de démarche. On touche ici à une problématique courante en analyse du contenu. En l'occurrence, doit-on songer à construire des systèmes de catégories --on dit des dictionnaires--, chacun approprié à des ensembles différents de situations, correspondant à différents aspects du monde, ou bien pourrait-on concevoir une sorte de dictionnaire universel, à la manière des catégories universelles de l'entendement humain de Kant?

Malheureusement, les questions qu'on rencontre en analyse du contenu ne sont pas toujours d'un tel intérêt théorique. Beaucoup de questions en effet sont simplement techniques, et d'autant contraignantes. La dépendance des systèmes d'analyse vis-à-vis de l'évolution de l'informatique est une de ces contraintes.

d. L'évolution de l'informatique.

Les systèmes d'analyse assistée par ordinateur sont toujours dépendants d'un certain environnement informatique et physique. Ainsi, le système WORDS, qui était supporté par le système OS (Operating System), n'est plus viable depuis que l'OS a été remplacé sur le site de BUCLLN11 par un système CMS-BATCH.

Avec la suppression de l'OS, on a dû en outre aménager le système GENERAL ANALYZER existant. Par la même occasion, on a amélioré les performances de l'ancien système, qui a en fait été entièrement réécrit, et on a créé de nouvelles possibilités d'exploitation des données dans un système appelé PROTAN. Mais on a fait en sorte que le système PROTAN intègre les possibilités de l'analyse catégorielle et celles de l'analyse non-catégorielle telles que le réalisait le système WORDS de Howard P. Iker.

II. Le système PROTAN comme logiciel

Le système PROTAN a d'abord été conçu comme logiciel, en ce sens que les données nécessaires à l'exécution d'une étape déterminée de l'analyse sont généralement créées par l'exécution en amont d'autres programmes qui créent ces données. Les programmes du système PROTAN sont ainsi interdépendants. Cette interdépendance n'empêche cependant pas de sortir du système en certains points de l'analyse, là où il peut être utile de créer des données, sous forme de fichier perforé, à traiter par un logiciel de statistique tel que le SAS³ par exemple. Le système PROTAN est en ce sens un système ouvert où les données générées par le système peuvent être analysées dans un autre système tel que le SAS par exemple, comme on vient de le signaler. En ce sens encore, le système WORDS serait un système fermé, les données qui y sont générées ne pouvant être traitées que par les programmes statistiques existant à l'intérieur du système.

Comme logiciel, le système PROTAN a d'abord été conçu comme un système d'analyse du contenu de type catégoriel. Il comporte donc tous les programmes de vérification de données, de tri de textes, et de comparaison de textes et dictionnaires, nécessaires à l'approche catégorielle, de même que les programmes de création et de gestion de dictionnaires. Mais le système PROTAN offre aussi la possibilité d'une approche non-catégorielle. Cette possibilité est réalisée par l'intermédiaire des programmes CWWCOL, CWORD, CWKWIC, CWKWOC, et surtout CWSELECT. On y reviendra dans un instant.

PROTAN présente en outre une certaine souplesse d'utilisation dans la mesure où le corpus peut y être divisé en plusieurs interviews, en plusieurs unités à l'intérieur des interviews, et en plusieurs locuteurs à l'intérieur des interviews et unités. Il revient à l'analyste de définir pour ses besoins propres la constitution des interviews, unités et locuteurs. Il est même possible, ayant inséré dans le texte des "marqueurs" dont le sens est déterminé par l'analyste, de resegmenter le corpus à partir de ces marqueurs; il est aussi possible de resegmenter le corpus par groupes de n phrases ou mots.

PROTAN permet enfin d'insérer des commentaires dans les fichiers de données, soit en tête de fichier, soit dans le texte lui-même. Ces commentaires ne sont jamais pris en compte dans le cours de l'analyse, mais les commentaires placés en tête de fichier accompagnent toutes les étapes de l'analyse. A ce titre, ils servent d'aide-mémoire dont l'usage est recommandé.

L'aspect non-catégoriel de l'analyse est réalisé par les étapes CWWCOL et CWORD d'une part, spécifiques à PROTAN ainsi qu'à l'ancien GENERAL ANALYZER, et par le programme CWSELECT d'autre part. (Les opérations de Key-Word-In-Context et de Key-Word-Out-of-Context, réalisées respectivement par les programmes CWKWIC et CWKWOC sont courantes et communes à presque tous les systèmes d'analyse de texte). Lorsqu'un corpus peut être divisé en deux sous-groupes selon une caractéristique quelconque, il peut être pertinent de chercher à savoir si la distribution de certains

³ La pratique consistant à sortir d'un système pour entrer dans un logiciel fort élaboré, tel que le SAS, est finalement plus efficace que celle consistant à construire, dans le système même, des programmes statistiques qui, de toute manière, n'atteindront jamais les performances et la souplesse des premiers. Voir Guillaumont et Minel (1986) pour un autre exemple.

mots ou groupes de mots, d'ailleurs laissés au choix ou à l'intuition de l'analyste, ne différencierait pas utilement les deux sous-ensembles ainsi constitués. Le programme CWORD effectue cette opération. Le programme CWCOL effectue une opération similaire, à la différence que les mots servant à la comparaison des distributions de fréquence ne sont pas vraiment choisis par l'analyste: Ils sont remplacés par un fichier de référence, qui n'est rien de plus qu'un autre texte, éventuellement constitué de la combinaison des textes issus des deux sous-ensembles. Ce que fait CWSELECT est une opération tout à la fois fort différente, beaucoup plus simple, et potentiellement plus puissante: CWSELECT crée, à partir d'un fichier de type "WORDS" créé par CSCUT, CRWSTRIP, ou CWEDIT une matrice de "mots par segment" où chaque case de la matrice représente la fréquence d'un mot dans un segment. Pour être complet sans entrer dans des détails qui font l'objet d'explications ultérieures, on ajoutera que le choix des mots qui forment l'abscisse de la matrice est laissé à l'utilisateur, et que les fréquences brutes des mots sont doublées de leur fréquences relatives par rapport à la somme des mots du texte. Cette matrice peut alors être introduite dans une grande variété d'analyses statistiques, multivariées, de clustering, ou autres.

Il y a enfin un troisième aspect de PROTAN qui lui est caractéristique, à côté des deux autres, catégoriel et non-catégoriel. C'est qu'il est possible de combiner une analyse par catégories avec une analyse par mots. Il est en effet possible de suivre au mot à mot la manière dont les mots d'un texte sont assignés à un système de catégories. Ceci offre deux avantages. D'abord, cette façon de procéder permet d'introduire les séquences de mots assignés et non assignés dans des analyses statistiques, soit de type séquentiel, soit de type "séries temporelles"; ensuite, cette façon de procéder permet de reconstituer un texte désarticulé dans son ordre séquentiel initial avec en outre les indications de mots assignés et non assignés. Les passages saturés dans une catégorie sont ainsi faciles à repérer.

Pour terminer, on voudrait enfin signaler que le format d'entrée des données du système PROTAN est le même que celui du système WORDS. Ceci eût été un avantage considérable si le système WORDS était resté viable à BUCLLN11, bien que, comme on vient de le souligner, PROTAN effectue à présent le même genre d'opérations sur les mots que ce que pouvait faire le système WORDS. Mais notre souci de rendre compatibles les formats d'entrée de deux systèmes différents signale un autre et dernier problème lié à l'analyse du contenu assistée par ordinateur, justement la variabilité des formats d'entrée des données d'un système à l'autre.

III. Description des programmes

A. Les noms des programmes

Les noms des programmes commencent tous par les lettres CS_, CD_, CR_, CF_, ou CW_. Ces lettres ont une signification qui aide l'utilisateur à se situer dans l'organisation du logiciel PROTAN.

La lettre 'S', comme deuxième lettre du nom du programme, indique que celui-ci effectue une opération sur le fichier-source (CSCHECK, CSSORT, CSEDIT, CSJOIN, et CSCUT), c'est-à-dire sur un fichier de type "SOURCE". La lettre 'D' indique, elle, une opération sur un fichier de type "DICTIONNAIRES" qui gère la création, l'édition ou l'utilisation de dictionnaires (CDCHECK, CDLISTA, CDLISTC, CDWLOOK et CDWJUXT). La lettre 'R' comme deuxième lettre du nom du programme indique que le programme traite, c'est-à-dire crée, édite ou utilise, un fichier de formes nominales et de radicaux de substantifs, adjectifs, et verbes (CRCHECK, CRLISTA, et CRWSTRIP), autrement dit, un fichier de type "ROOTS". La lettre 'F' comme deuxième lettre d'un nom de programme signale une interaction avec un fichier de type "FINDS" (CFCHECK, CFLISTA, CFWKWIC); ces derniers fichiers ont en commun de viser à modifier ou à supprimer des occurrences de certains mots ou racines à des adresses précises. Enfin, la lettre 'W' indique que le programme effectue un traitement sur les mots⁴ (words), c'est-à-dire sur un fichier de type "WORDS". Différents traitements sur les mots (ou lexies) sont possibles. Ces traitements peuvent avoir pour objet la comparaison de fréquences de mots (CWWORD et CWWCOL), ou le tri des mots en ordre de fréquence inverse (CWTALLY), ou d'une sélection de mots (CWSELECT); ou ils peuvent avoir pour objet la comparaison d'un texte avec un ou des dictionnaire de mots assignés à des catégories (CDWLOOK et CDWJUXT), ou encore la comparaison d'un texte avec un dictionnaire de lexies assignées à des formes nominales, lexèmes, formes libres et radicaux (CRWSTRIP). Le programme CFWKWIC a, lui, pour objet la préparation des fichiers utilisés par certains des programmes qui exécutent des traitements spécifiques sur les mots.

Par ailleurs, certains programmes du logiciel PROTAN créent aussi un fichier-système utilisable par d'autres programmes. On reconnaît ces fichiers-système (WORDS, DICTIONNAIRES, FINDS, et ROOTS) au fait qu'ils se terminent par la lettre 's', pour 'système'.

B. Résumé des programmes

En quelques mots à présent, et sans entrer dans des détails qui sont présentés ultérieurement dans l'analyse approfondie de chaque programme, voyons ce que font les différents programmes qui composent le logiciel PROTAN.

1. CDCHECK: Ce programme crée le fichier contenant les mots du dictionnaire de catégories qui sera utilisé par CDWLOOK.

2. CDLISTA/CDLISTC: Ces deux programmes impriment le fichier créé par CDCHECK, soit en ordre alphabétique (CDLISTA), soit selon l'ordre des catégories (CDLISTC).

⁴ Il faudrait parler de lexies plutôt que de mots, le mot n'étant pas une unité linguistique reconnue.

3. CDWJUXT: Ce programme compte les co-occurrences de deux catégories ou séries de catégories à l'intérieur d'une même phrase et fait le total des co-occurrences par segment.

4. CDWLOOK: Ce programme compare les mots du texte aux mots contenus dans un dictionnaire de catégories ou de normes, en effectuant les comptages correspondants.

5. CFCHECK: Ce programme crée le fichier-système de type "FINDS" qui permet la recherche d'adresses d'occurrences de mots.

6. CFLISTA: Ce programme imprime le fichier de type "FINDS" créé par le programme CFCHECK.

7. CFWKWIC: Ce programme imprime le contexte des occurrences de mots particuliers qu'on cherche à modifier ou à supprimer, et prépare le fichier nécessaire pour exécuter ces modifications.

8. CRCHECK: Ce programme crée le fichier contenant d'une part les lexies et d'autre part leurs formes nominales ou autres unités minimales.

9. CRLISTA: Ce programme imprime le fichier créé par CRCHECK, en ordre alphabétique, soit sur les lexies, soit sur les unités minimales.

10. CRWSTRIP: Ce programme permet de transformer, à l'aide d'un ou plusieurs dictionnaires⁵ appropriés, les lexies en leurs formes nominales; il permet aussi d'en supprimer. Il a spécifiquement pour but de ramener les lexies à des unités minimales (lexèmes, radicaux, formes libres).

11. CSCHECK: Le premier programme du logiciel vérifie si les données --le texte et les informations codées d'interviews, d'unités, et de locuteurs-- ont été "entrées" correctement dans leurs colonnes respectives.

⁵ On parlera dans la suite du texte de "dictionnaires de formes nominales" pour désigner les "dictionnaires de lexies assignées à des formes nominales, lexèmes, formes libres, et radicaux", qui est une formulation correcte quoiqu'un peu longue...

Selon le Dictionnaire de la linguistique, de Mounin, "les lexies sont les unités de surface du lexique, les entrées du dictionnaire, qui comprennent les lexèmes, leurs dérivés affixaux et les composés. Pomme, pommier, pomme de terre sont alors des lexies, alors que seul pomm(e) est un lexème".

A ce même propos, le terme de forme libre désigne, en linguistique,

"une forme linguistique pouvant à elle seule constituer un énoncé, par opposition à la forme liée qui ne le peut pas: utilement résulte de la combinaison de la forme libre utile et de la forme liée -ment".

Enfin, on appellera radical

"l'élément commun à une famille de mots: livr- dans livrer et livraison".

12. CSCUT: Ce programme est le point de passage obligé de toutes les analyses qu'on peut faire avec le système PROTAN. CSCUT crée une liste alphabétique des mots du texte avec leur fréquence tout en créant un fichier-système, de type "WORDS" nécessaire aux autres programmes.

13. CSEDIT: Il s'agit ici encore d'un programme d'édition qui permet de modifier le contenu des informations contenues dans les colonnes 73 à 80. (Il n'y a pas, à proprement parler, dans PROTAN, de programme d'édition du texte contenu dans les colonnes 1 à 70, dans la mesure où une telle édition se fait aisément en CMS).

14. CSJOIN: Ce programme d'édition, avec CSSORT et CSEDIT, permet de mettre bout à bout, sous un seul nom de fichier, plusieurs textes enregistrés sous des noms différents.

15. CSSORT: Ce programme d'édition permet, si on le désire, de trier, non pas le texte, mais les informations codées contenues dans les colonnes 73 à 80. Ce programme ne fait que trier ces informations sans en modifier le contenu.

16. CWADD: Ce programme permet d'insérer, à différents endroits d'un fichier de type "WORDS", des informations non textuelles qui peuvent néanmoins être comptabilisées dans les calculs de fréquences et tout ce qui s'y rapporte. Sont concernés par ce programme toutes les variables paralinguistiques, comportementales, non verbales, et physiologiques, comme le rythme cardiaque par exemple.

17. CWEDIT: Ce programme effectue les modifications et suppressions de mots à des adresses particulières selon le fichier préparé par le programme CFWKWIC.

18. CWFLOW: donne la moyenne mobile du nombre de mots nouveaux introduits dans des intervalles de textes successifs.

19. CWKWIC: Ce programme permet de trier certains mots du texte en les maintenant dans leur contexte (KWIC pour Key-Word-In-Context).

20. CWKWOC: Ce programme a pour but d'extraire du corpus un ensemble de mots à partir de certains mots-clés. (KWOC pour Key-Word-Out-of-Context.)

21. CWNEW: identifie et dénombre les mots nouveaux qui apparaissent dans un texte par rapport à un ou plusieurs textes antérieurs.

22. CWPAT: recherche et identifie, à l'aide d'opérateurs logiques booléens, des patterns de mots éloignés les uns des autres.

23. CWREFER: Ce programme sélectionne (sur la base d'un critère de fréquence minimale) et crée, de manière automatique pour l'utilisateur, une liste de mots dans un format de type "REFER" directement utilisable dans le programme CWSELECT.

24. CWSELECT: Ce programme crée, à partir d'une liste de mots fournis par l'utilisateur, une matrice de "mots par

segment"; chaque case de la matrice représente la fréquence d'un mot dans un segment.

25. CWTALLY: Ce programme effectue un tri des mots du texte soit sur leur fréquence, soit sur leur longueur.

26. CWWCOL: Programme permettant de comparer deux séries de textes, soit une série dite "expérimentale" et une série dite "de contrôle", par rapport à un texte dit "de référence", et d'identifier les mots qui différencient statistiquement les deux séries de textes.

27. CWORD: Comme le précédent, ce programme compare deux séries de textes, mais cette fois, la comparaison se fait en référence à une liste de mots choisis par l'analyste en fonction de ses hypothèses.

C. La codification des messages

Pendant l'exécution du travail et à l'issue de l'exécution, PROTAN émet un certain nombre de messages. Certains de ces messages sont propres à chaque programme dont ils reprennent généralement dans leur code les quatre premières lettres, soit CSCH....pour les messages émis par CSCHECK, CSCU....pour CSCUT, etc. Les messages généraux du système sont repris en fin de cette section, tandis que les messages spécifiques émis par chaque programme figurent à la fin de la présentation détaillée de chaque programme.

Ces messages peuvent être de simples informations, signalant par exemple la fin normale de l'exécution d'un travail. Ces messages peuvent également être des avertissements signalant certaines particularités du texte soumis à l'analyse. Par exemple, dans le programme CSCHECK, un mot de plus de 20 caractères sera signalé par le message "CSCH373W ITEM WILL BE TRUNCATED" avec en outre le signe "\$" juste en dessous du mot qui fait l'objet du message. Les messages d'avertissement ne constituent pas des erreurs et n'entraînent jamais l'arrêt du programme. Cependant, un message d'avertissement doit être pris en compte par l'utilisateur car il peut révéler une erreur réelle: si le mot "anticonstitutionnellement" (25 caractères, dont les 5 derniers, "ement", seront tronqués) ne constitue pas une erreur, la chaîne de caractères "messaged'avertissement", de 21 caractères, en est une évidente.

Les messages d'information se reconnaissent à la présence de la lettre I comme dernière lettre du code du message, par exemple "CSCH151I USER'S COMMENT", qui, dans le programme CSCHECK, signifie "Cette ligne est un commentaire inséré par l'utilisateur dans le texte sans en faire partie". Les messages d'avertissement se reconnaissent, eux, à la lettre W, pour "warning", comme dernière lettre du code du message. Par exemple, "CDCH224W CATEGORY XXX IS NEVER REFERENCED" qui, dans le programme CDCHECK dont la fonction est de créer un dictionnaire de catégories, signifie "Vous avez introduit une catégorie libellée XXX (numéro de catégorie), mais le programme CDCHECK n'a pas trouvé de mot qui soit attribué à cette catégorie".

Une dernière série de messages est constituée des messages d'erreur. Par exemple, le programme CSCHECK émet le message "CSCH383E INT SEQUENCE ERROR" si, disons l'interview 004, dans les colonnes 73 à 75, est suivi, à un certain endroit du texte, de l'interview 002, ce qui signale une erreur dans la séquence des

interviews. Les messages d'erreur qui se terminent par la lettre E n'entraînent pas l'arrêt du programme. Au contraire, le programme, quel qu'il soit, poursuit son exécution, envers et contre tout, tout en relevant au passage les erreurs qu'il rencontre.

D'autres messages d'erreur se terminent par les lettres S ou U, respectivement pour "severe error" et "unrecoverable error". Il s'agit ici d'erreurs telles qu'elles entraînent très vite l'arrêt du programme qui, pour ainsi dire, "n'est pas capable de retomber sur ses pieds" en présence de pareille erreur. On aura ainsi une erreur de type U lorsque, dans le programme CRCHECK créant le dictionnaire qui assigne les lexies à leurs unités minimales, le programme découvre qu'il y a plus de 99.999 mots à garder en mémoire pour l'opération ultérieure de nominalisation/-radicalisation que devra exécuter le programme CRWSTRIP (CRCH304U USER'S STRIPLIST LENGTH > 99999). On notera encore que, dans certains cas d'erreurs de type S ou U, entraînant l'arrêt du programme, l'utilisateur devra consulter en outre le fichier-console, c'est-à-dire le fichier qui détaille toutes les opérations logiques qu'a effectuées le programme, pour connaître les raisons de la présence du message d'erreur de type S ou U. De toute façon, erreur ou non, il est bon de savoir aussi que le fichier-console contient un code de retour de l'exécution du programme (RC=X, abrégé de "return code") qui doit être égal à 0 dans le cas d'une fin normale de travail. Le code de retour sera de 4 dans le cas de messages d'avertissement, de 8, 12, ou 16 dans le cas de messages d'erreurs de type E, S, ou U respectivement. On notera enfin que les exécutions qui ont entraîné un diagnostic d'erreur S ou U ne produisent pas de fichiers utilisables.

Quant aux messages généraux du système, ils se subdivisent en neuf classes de messages dont le détail suit.

1. Fin d'exécution des programmes

*CYEX001I	processing terminated - normal end of job
*CYEX002W	processing terminated - however "warning" diagnostics have been generated
*CYEX003E	processing incomplete - "error" diagnostics have been generated
*CYEX004S	processing failed - "severe error" diagnostics have been generated
*CYEX005U	processing stopped - an "unrecoverable error" diagnostic has been generated

2. Détection d'une fin de fichier

*Cxxx022U	empty data set on 'xxxxxxxx'
*Cxxx023I	end of data set reached on 'xxxxxxxx' - lines read from nnnnn to nnnnn - try to open the next data set if any
*Cxxx029S	end of data set reached on 'xxxxxxxx' - xxxxxxxx
*Cxxx032I	end of data set reached on 'xxxxxxxx' - xxxxxxxx
*Cxxx034U	premature end-of-file on 'xxxxxxxx'

3. Incapacité d'ouvrir un fichier

```
*Cxxx053U ddname or file 'xxxxxxx' missing
*Cxxx061S end of data set chain reached when opening
          'xxxxxxx' - no xxxxxxxx record has been read
*Cxxx062I end of data set chain reached when opening
          'xxxxxxx' - xxxxxxxx file complete
*Cxxx063S end of data set chain reached when opening
          'xxxxxxx' - no word has been read
*Cxxx064S end of data set chain reached when opening
          'xxxxxxx' - stopped in general comments sequence
*Cxxx070E cannot open 'xxxxxxx' - xxxxxxxx
*Cxxx071S cannot open 'xxxxxxx'
*Cxxx072U cannot open 'xxxxxxx'
```

4. Allocation dynamique de mémoire

```
*Cxxx115E insufficient storage available for 'xxxxxxx':
          nnnnnn bytes needed - xxxxxxxx
*Cxxx117S insufficient storage available for 'xxxxxxx':
          nnnnnn bytes needed
*Cxxx118I insufficient storage available for 'xxxxxxx':
          nnnnnn bytes needed
```

5. Options

```
*CYPL124S page length must be greater than 20
*Cxxx125S option value 'xxxxxxx' has been truncated
*Cxxx126S option syntax error 'xxxxxxx'
```

6. Mots-clés, identifications, et commentaires

```
*Cxxx143S identification does not begin in col. 9
*Cxxx144S blank imbedded into identification
*Cxxx145S invalid character in identification
*Cxxx148S more then 999 general comment lines
*Cxxx149I end of general comments
*Cxxx150I 999th comment dropped
*Cxxx151I user's comment
*Cxxx155E invalid character into word/keyword
*Cxxx158W word/keyword will be truncated
*Cxxx160S no search word/keyword matched
*Cxxx161S word/keyword list greater than nnn
*Cxxx162S this is not a 'xxxxxxx' file
```

7. Génération incomplète de fichier-système

```
*Cxxx171U  'xxxxxxx' file unsuccessfully processed by
           'xxxxxxx'
*Cxxx172S  'WORDS' file may not be processed more than 255
           times by "CWEDIT" or "CRWSTRIP"
```

8. Erreur de transmission lors d'une entrée/sortie

```
*Cxxx184S  input/output error on 'xxxxxxx'
```

D. Note sur les procédures d'appel des programmes⁶

L'appel général des programmes de PROTAN, explicité par l'utilisateur ou implicite, quelque soit l'environnement, se présente sous la forme d'une ligne de commande constituée du nom du programme suivi d'une liste d'options dans un ordre quelconque, soit:

```
nom_du_programme slisting = 'chemin_et_nom_du_fichier_listing',
                    smaster =
'chemin_et_nom_du_fichier_de_paramètres_et_options',
                    pl = nombre_de_lignes_par_page_de_listing;
```

1. En DOS

a. Appel via la commande batch "**protan.bat**":

En DOS (et en UNIX, voir point 2), l'appel s'effectue de manière conviviale et simple (PATHs fixés avant chaque exécution, mention du niveau d'erreur, etc., inclus dans la commande globale), soit:

```
call protan.bat nom_du_programme <nom_du_fichier_listing
                                <nom_du_fichier_de_paramètres_et_options
                                <nombre_de_lignes_par_page_de_listing>>>
```

Notons que "*call protan.bat*" est valable s'il est inclus dans un fichier ".bat". En cas d'appel direct, il ne faut pas mentionner "call". L'encadré qui suit illustre ce que pourrait contenir un tel fichier batch "**Protan.bat**":

⁶ Tous les exemples de ce manuel ont été conçus pour un environnement DOS. La présente note présente l'appel général des programmes de PROTAN dans les environnements DOS et UNIX. Une note séparée présente l'appel des programmes dans l'environnement VM/CMS.

```
@echo off
set oldpath=%path%
path d:\protan
: PATH à adapter à chaque configuration informatique
if x%1==x? goto explain
if x%1==x goto no1
if x%2==x goto no2
if x%3==x goto no3
if x%4==x goto no4
%1 slisting='%2',smaster='%3',pl=%4;
goto job_terminated
:explain
echo syntax of PROTAN.BAT :
echo PROTAN program_name [listing_file [opt&par_file [pagesize] ]
]
goto exit1;
:no1
echo program name missing.
goto explain
:no2
%1
goto job_terminated
:no3
%1 slisting='%2';
goto job_terminated
:no4
%1 slisting='%2',smaster='%3';
:job_terminated
if errorlevel 17 goto rcunknown
if errorlevel 16 goto rc16
if errorlevel 12 goto rc12
if errorlevel 8 goto rc8
if errorlevel 4 goto rc4
goto exit
:rcunknown
echo unknown error condition; see PROTAN's responsible.
goto exit
:rc16
echo an UNRECOVERABLE ERROR diagnostic has been generated.
goto exit
:rc12
echo at least one SEVERE ERROR diagnostic has been generated.
goto exit
:rc8
echo at least one ERROR diagnostic has been generated.
goto exit
:rc4
echo at least one WARNING diagnostic has been generated.
:exit
echo suppression of temporary files
erase *.tmp
:exit1
path %oldpath%
set oldpath=
echo on
```

b. Appel de programmes "en chaîne"

Il est possible, en DOS comme en UNIX, d'appeler en chaîne plusieurs programmes via un fichier d'appels multiples. Par exemple, le fichier "cscut1.bat":

Exemple d'appel de programmes de PROTAN en DOS

```
call protan.bat cscut camusscu.lis camusscu.scu
if errorlevel 12 goto exit
: si le code de retour est >= 12, on arrête
: sinon on exécute le programme suivant

call protan.bat crwstrip camusrws.lis camusrws.rws 60

: et ainsi de suite

:exit
```

2. En UNIXa. Appel via la commande shell "**protan**"

L'appel des programmes en UNIX est semblable à l'appel en DOS, sauf que l'expression "call" dans `<call protan.bat ...>` disparaît et que ce qui s'appelait en DOS `<protan.bat>` s'appelle en UNIX `<protan>` tout court (en minuscules!). Ce fichier `<protan>` doit avoir les permissions d'un fichier exécutable (`chmod a+x protan`).

L'encadré suivant (fonctionnellement identique au fichier **protan.bat** qui précède au point 1) en illustre le contenu:

```

#!/bin/sh
#set -x
oldpath=$PATH
PATH=./home/upso/hogenraa/protan
# PATH à adapter à chaque configuration
case $# in
  0) echo program_name missing.
     rc=20;;
  1) if test $1 = '?'
     then rc=2
     else $1
        rc=$?
     fi;;
  2) $1 slisting='\$2\'';
     rc=$?;;
  3) $1 slisting='\$2\'' , smaster='\$3\'';
     rc=$?;;
  4) $1 slisting='\$2\'' , smaster='\$3\'' , pl=$4;
     rc=$?;;
  *) echo too many arguments.
     rc=20;;
esac
if test $rc -eq 1
then rc=20
fi
case $rc in
  0) ;;
  2|20) echo syntax of protan:
        echo protan program_name [listing_file [opt\&par_file [page_size] ] ];;
  4) echo at least one WARNING diagnostic has been generated.;;
  8) echo at least one ERROR diagnostic has been generated.;;
  12) echo at least one SEVERE ERROR diagnostic has been generated.;;
  16) echo an UNRECOVERABLE ERROR diagnostic has been generated.;;
  *) echo unknown error condition\; see protan\'s responsible.;;
esac
if test $rc -ne 2 -a $rc -ne 20
then if test -f *.tmp
    then echo suppression of temporary files.
        /usr/bin/rm *.tmp
    fi
fi
PATH=$oldpath
oldpath=
exit $rc

```

b. Appel de programmes "en chaîne" (UNIX)

Il est également possible d'appeler plusieurs programmes "en chaîne" par un fichier d'appels multiples. Par exemple, le fichier shell "*CSCUT1*" doit avoir la permission d'un fichier exécutable, soit (`chmod a+x cscut1`):

Exemple d'appel de programmes de PROTAN en UNIX

```
#!/bin/sh
set -x
protan cscut camusscu.lis camusscu.scu
rc=$?

# si le code de retour est > 8, on arrête
if test $rc -gt 8
then exit $rc
fi
# sinon on exécute le programme suivant

protan crwstrip camusrws.lis camusrws.rws 60
rc=$?

# et ainsi de suite

exit $rc
```

3. En Macintosh

On part ici du présupposé que les programmes de PROTAN se trouvent dans un dossier spécifique qu'on pourrait appeler le dossier PROTAN. Pour invoquer et exécuter un des programmes de PROTAN, il faut:

- a. faire un double-clic sur l'icône correspondant au programme qu'on veut exécuter, et
- b. répondre aux trois questions qui apparaissent dans la fenêtre, en précisant:

(1) chemin et nom du listing: il s'agit du listing sur lequel seront écrits les résultats du programme exécuté ainsi que les divers messages produits à cette occasion. La façon d'écrire un chemin en Macintosh est expliquée plus loin. Un nom de chemin valide est par exemple:

"disque dur:hamlet:hamlet.cscheck.listing"

(sans les guillemets). Faire **ENTER** pour enregistrer ce nom de listing.

(2) chemin et nom de fichier de paramètres et options: voir plus loin. Un nom de fichier valide est par exemple:

"disque dur:hamlet:hamlet.cscheck"

(sans les guillemets). Faire **ENTER** pour enregistrer ce nom de fichier.

(3) éventuellement le nombre de lignes par page sur le listing, 60 étant le nombre par défaut. Faire **ENTER** même si on choisit l'option par défaut. Ce dernier **ENTER** déclenche l'exécution complète du programme.

(4) La fin de l'exécution du programme est signalée par l'apparition d'un message précisant un code de retour

(précédé éventuellement d'un message d'erreur grave). Faire **ENTER** ou cliquer sur **QUIT** du menu **FILE** pour fermer le programme et les différents fichiers.

c. Chemins et noms de fichiers en Macintosh.

Les nom de fichiers peuvent avoir jusque 31 caractères en Macintosh, y compris des blancs. Les fichiers font toujours partie d'un dossier (à moins d'être rattachés directement à la racine du disque dur ou d'une disquette). Noms de fichiers et dossiers sont séparés par le signe ":".

Le dossier où se trouvent les programmes de PROTAN sera appelé ici le dossier courant. Chaque fichier peut être appelé soit de manière relative, soit de manière absolue:

(1) L'appel "*absolu*" se fait à partir du nom du disque suivi des différents dossiers intermédiaires à ouvrir avant d'atteindre le fichier. Tous les éléments d'un chemin sont enchaînés par un ":". Un exemple d'appel absolu est le suivant:

disque dur:protan:analyse:ern:ern.cdwjuxt.liste

(2) L'appel "*relatif*" se fait à partir du répertoire (=dossier) courant et doit alors commencer par un ":". Un appel relatif serait par exemple:

:analyse:ern:ern.cscut

d. Particularités de PROTAN en Macintosh

A l'aide votre éditeur, veillez à ne jamais écrire des lignes de plus de 80 caractères dans les fichiers. A retenir aussi qu'il n'est pas possible ni d'automatiser ni d'enchaîner les programmes. Pour les programmes qui demandent des listes de noms de fichiers après la liste d'options (c'est-à-dire après le point-virgule), comme dans CSCHECK, CSCUT, etc., il FAUT aligner les noms des fichiers (et de leurs chemins) les uns EN DESSOUS des autres, à partir de la première colonne.

4. Remarques générales concernant les applications de PROTAN en DOS, UNIX, et Macintosh

Pour connaître les valeurs par défaut de toutes les options "*s...*", y compris "*slisting*" et "*smaster*", se référer aux premières pages du listing produit par exemple par la commande:

"protan.bat cscut" ou "call protan.bat cscut".

L'option "*stemp*" (nom générique des fichiers temporaires) est utilisée dans tous les programmes excepté dans CSCHECK, CSEDIT, CSJOIN, CWREFER, CFLISTA, CPEXCOR, CDLISTA, et CDLISTC. La syntaxe de "*stemp*" est <stemp = '*chemin éventuel suivi de *.tmp*'> (ou autre extension), soit, par exemple:

```
"*.tmp" (valeur par défaut dans le répertoire courant7)  
      (valable également en UNIX)
```

```
ou
```

```
"c:\temp\*.trh"
```

Si l'analyste décide par exemple de mettre les fichiers temporaires de travail dans le répertoire "c:\protan\tests\laval*.tmp", ces fichiers ne seront détruits que si la procédure *protan.bat* est modifiée en conséquence. En l'occurrence, le "erase *.tmp" à la fin de la procédure *protan.bat* doit être modifié en :

```
"erase c:\protan\tests\laval\*.tmp".
```

Dans la procédure *protan* en UNIX, on aura par exemple :

```
"rm /home/protan/tests/laval/*.tmp"
```

Les valeurs alphanumériques demandées par les options sont encadrées d'apostrophes, soit "xyz = 'xxx'" (par exemple, dans le programme CSCUT, <SEGT = 's'>. Mais les valeurs numériques demandées par les options ne le sont pas, soit "abc = 999" (par exemple, dans le programme CSCUT, <CTRL = 20>.

Dans le fichier correspondant à "*smaster*", les valeurs '*' assignées à certaines options annoncent une liste de noms de fichiers (sources, etc...) qu'il faudra placer en un endroit convenu après la liste d'options (voir l'abrégé pour plus de détails).

⁷ Adapter le fichier **protan.bat** (ou le fichier **protan** en UNIX) si on modifie les attributs de cette valeur par défaut.

IV. Les fichiers de données du système PROTAN et leur emploi

Une analyse porte nécessairement sur quelque chose. Ce "quelque chose" peut être des observations, des chiffres, des fragments de matière, etc. En analyse du contenu, ce quelque chose, qu'on appelle les données, est verbal. Les données se présentent en effet sous la forme d'un texte continu; certaines données se présentent aussi sous forme de listes de mots codés. Par "fichier de données", on entendra donc le texte qu'on souhaite analyser, on entendra aussi tout matériel verbal permettant de faire progresser cette analyse.

Le système PROTAN utilise quatre types de fichiers de données. Il y a d'abord les fichiers de données SOURCE, utilisés par tous les programmes dont le nom commence par les lettres CS-, soit CSCHECK, CSJOIN, CSEDIT, CSSORT et CSCUT. Il y a ensuite les fichiers de données INROOT, utilisés par le programme CRCHECK; ces fichiers INROOT peuvent être générés manuellement par l'utilisateur, ils peuvent aussi être créés de façon semi-automatique par le programme CWKWOC. Et il y a enfin les fichiers de données INDICT utilisés par le programme CDCHECK et les fichiers de données INFIND utilisés par le programme CFCHECK pour la recherche d'adresses d'occurrences de mots.

Pour clarifier les termes. On parlera dans la suite du manuel, de fichier-source pour désigner l'ensemble du corpus à traiter au cours d'une même analyse; ce fichier doit toujours être précédé des symboles syntaxiques qui le caractérisent. Mais ce fichier-source peut lui-même être divisé, et en n'importe quel endroit, en autant de fichiers CMS distincts qu'il est souhaitable (99 au maximum), et on parlera alors de fichiers partiels. Un principe identique régit les dictionnaires de formes nominales (fichiers "INROOT"), les dictionnaires de normes et de catégories (fichiers "INDICT"), et les fichiers de recherche d'adresses d'occurrences de mots (INFIND).

Avant d'examiner séparément chacun de ces fichiers de données, il faut relever deux conventions d'écriture qui les concernent. A propos des noms propres d'abord, il y a la convention, arbitraire mais utile, qui veut que ceux-ci soient toujours précédés du signe \$: lors d'un tri en ordre alphabétique, les noms propres figurent ainsi en tête de liste, "\$JEAN", "\$USA"⁸, etc... De plus, comme le système PROTAN ne distingue pas les majuscules des minuscules, certains noms propres pourraient perdre leur qualité, comme par exemple "PIERRE" qui, ainsi écrit, peut être matière ou prénom masculin. "\$PIERRE" écarte toute ambiguïté sous ce rapport. Ensuite, à propos des accents propres à la langue française, il y a la convention selon laquelle les accents aigus, graves, et circonflexes, et les trémas sont remplacés respectivement par les chiffres 1, 2, 3, et 4 figurant à la suite de la voyelle accentuée; les signes diacritiques analogues (à, où) obéissent à la même convention; enfin, le ç (c cédille) s'écrit "c1". Par exemple, la phrase:

"Les élèves français rêvent aux vacances de Noël"

⁸ L'exemple "USA", qui s'écrit normalement, ou souvent, "U.S.A.", est l'occasion de remarquer qu'il ne serait pas souhaitable d'écrire "\$U.S.A." dans un fichier-source, au risque de créer trois phrases différentes, composées de "\$U", "S", et "A". On veillera donc à n'utiliser les points et autres séparateurs que si on souhaite vraiment créer des mots ou des phrases distinctes.

deviendra "LES E1LE2VES FRANClAIS RE3VENT AUX VACANCES DE \$NOE4L". Les différents dictionnaires en langue française qui sont présentés au chapitre 6 respectent ces conventions.

E. Les fichiers de données SOURCE

Le fichier-source est constitué de deux parties qui figurent côte à côte, à savoir les textes à analyser d'abord, et les codes de segmentation ensuite. Mais avant même la partie "texte" et la partie "codes de segmentation", un fichier-source s'annonce par trois types d'enregistrements dont le deuxième est d'ailleurs facultatif:

a. Le premier enregistrement est constitué de:

```
*SOURCE-iiiiiiiiccccccccc.....c
```

(1) où iiiiiiic est à remplacer par un nom de 1 à 8 caractères alignés à gauche. Ce nom permet d'identifier le corpus tout au cours des analyses subséquentes. Ces caractères peuvent être les lettres de A à Z, les chiffres de 0 à 9, ainsi que les caractères (entre < et >, non compris) <+ & \$ * % _ @ ->.

Le choix du nom n'a en soi pas beaucoup d'importance, sinon qu'il doit permettre de reconnaître sans ambiguïté le matériel verbal en cours d'analyse. Mais il est utile de savoir que ce nom va accompagner les diverses transformations et réductions que va connaître le texte. A ce titre, le choix d'un nom significatif est de nature à faciliter l'analyse.

(2) où ccc...c est à remplacer par un commentaire, facultatif, qui peut occuper les colonnes 17 à 80.

b. Le deuxième type d'enregistrement, d'ailleurs facultatif comme on l'a dit plus haut, est constitué de commentaires qui peuvent occuper les colonnes 1 à 80 sur un maximum de 999 lignes. S'il n'est peut-être pas nécessaire d'insérer jusque 999 lignes de commentaires en tête d'un texte, il peut par contre être fort utile d'insérer diverses informations concernant l'origine du texte, la sélection éventuelle de textes à laquelle on a procédé, ou encore le rappel des passages de textes qui auraient été ignorés, le ou les auteurs du texte, l'édition particulière d'où a été tiré le texte, ou encore les noms des auteurs de l'analyse, avec la référence du crédit qui a éventuellement soutenu le travail d'analyse. Bref, il y a autant de commentaires possibles que de situations différentes, la règle étant que la trace sur le papier reste, alors que "la mémoire, elle, fout le camp", comme on nous apprend à ne pas dire --mais même cela s'oublie--.

c. Le troisième type d'enregistrement, obligatoire et unique, comme le premier, est constitué de:

```
*ENDCOM~
```

débutant en colonne 1 et qui annonce la fin de la zone de commentaires. Cet enregistrement doit toujours être présent, même si aucun commentaire n'a été inséré.

d. La partie "texte" du fichier-source.

Vient ensuite le corpus proprement dit, à l'analyse duquel on veut procéder. On peut utiliser différents jeux de caractères

pour créer le texte. Certains de ces caractères peuvent entrer dans la composition d'un mot, d'autres non.

Ainsi, les caractères A à Z, 0 à 9, et (entre < et > non compris) <+ & \$ * % _ @> peuvent entrer dans la composition d'un mot, de même que le <-> s'il n'est pas redoublé et s'il ne figure pas en début ou en fin de mot. Par contre, les autres caractères (entre < et > non compris) <~ < (|) ~ / , > ' = " > vont jouer comme séparateurs de mots, de même que le double ou le triple tiret <-->, <--->, etc...

Par exemple, "mot-clé" fait une chaîne de caractères, alors que "mot--clé" équivaut à "mot" et à "clé", soit deux chaînes de caractères. Ainsi encore de "c'est", qui devient "c" et "est", soit deux chaînes, l'apostrophe jouant comme séparateur.

On a alors d'autres séparateurs qui, eux, séparent les phrases. Les caractères <! . ; ? :> sont les cinq séparateurs de phrases qui opèrent par défaut dans le système PROTAN. Ceci veut dire qu'en l'absence d'autres instructions, le système crée, et compte, une nouvelle phrase chaque fois qu'un de ces caractères est rencontré, tout comme le système crée, et compte, un autre mot chaque fois qu'un séparateur de mot est rencontré.

Il est cependant possible d'imposer au système PROTAN le ou les caractères séparateurs de phrases de son choix (voir l'option SENT des programmes CSCHECK ou CSCUT). On peut alors choisir les caractères séparateurs de phrases parmi la liste des caractères opérant par défaut et parmi la liste des caractères séparateurs de mots, à l'exception cependant du caractère blanc (~).

Dans un texte écrit, la ponctuation est fixée à l'avance par l'auteur du texte; il n'y a qu'à la retranscrire fidèlement. Il se pose quelques difficultés dans le cas d'un texte enregistré, sur bande magnétique par exemple, retranscrit par l'utilisateur. Il n'est pas toujours aisé dans ce cas de reconnaître si, en un point donné du texte, on est en présence de la fin d'une phrase --demandant un séparateur de phrase-- ou de la fin d'une proposition --pour laquelle une virgule suffirait--. Ces questions ne sont pas tout à fait étrangères à l'analyse du contenu, et au système PROTAN en particulier, dans la mesure où ce système permet justement de "resegmenter" un texte en groupes de n phrases.

Cette dernière remarque nous conduit à parler du dernier caractère possible dans un texte, à savoir le caractère <#> qui, entourant une chaîne de caractères⁹ dont la signification est déterminée par l'utilisateur, transforme cette chaîne particulière en un "marqueur", par exemple

"#T20#".

L'usage du marqueur est en fait encore une autre manière de resegmenter un texte, et on se référera pour ce sujet au programme CSCUT. On pourrait d'ailleurs généraliser cette dernière remarque en disant que l'usage de l'option SENT dans les programmes CSCHECK et CSCUT est aussi une manière de resegmenter un texte.

Ceci étant, le texte, avec ses caractères composant des mots et ses caractères séparateurs, doit se placer entre les colonnes 1 à 70, les colonnes 71 et 72 restant vides, et les colonnes 73 à 80 étant réservées aux codes de segmentation, la colonne 76 restant vide également.

Avant d'en venir là, il faut encore signaler que le système PROTAN accepte des mots qui ont jusqu'à 20 caractères de long, les

⁹ De 20 caractères au plus, caractère <#> compris.

caractères à partir du vingt-et-unième étant tronqués. LES MOTS NE PEUVENT PAS ETRE COUPES.

La dernière remarque de ce point consacré à la partie "texte" d'un fichier-source, concerne les commentaires. On connaît maintenant les commentaires qui, des colonnes 17 à 80, peuvent suivre l'identification du fichier-source, et on connaît aussi les commentaires qui peuvent suivre cette ligne d'identification jusqu'à l'enregistrement "*ENDCOM~".

Il y a un dernier type de commentaire qui peut, lui, s'insérer directement dans le texte. Il suffit pour cela, à l'endroit du texte où on souhaite insérer la ou les lignes de commentaires, de ne pas mentionner les trois codes de segmentation prévus entre les colonnes 73 à 80.

Ces commentaires ne feront pas partie des analyses à suivre; placés comme ils le sont dans le texte, ils n'accompagneront pas les diverses transformations du fichier-source (à la différence des autres types de commentaires qui, eux, suivent les transformations du fichier-source sans être modifiés eux-mêmes).

Il peut être utile d'utiliser de tels commentaires à l'intérieur du texte afin qu'au moins au niveau du fichier de données de base, il reste une trace des remarques concernant l'un ou l'autre aspect du texte, de son édition, de suppressions éventuelles de matériel, etc..., bref, de toute observation qu'on peut faire au cours de l'opération de "rentrée des données".

e. La partie "codes de segmentation" d'un fichier-source.

Le système PROTAN prévoit trois codes de segmentation, qui sont le code d'interview (code I, entre les colonnes 73 à 75), le code d'unité (code U, entre les colonnes 77 à 79), et le code de locuteur (code S, en colonne 80). Comme on l'a indiqué précédemment, l'absence des trois champs de segmentation transforme le texte en un commentaire échappant à l'analyse.

Le code I et le code U ont un format libre, c'est-à-dire qu'ils ne doivent pas nécessairement être alignés à droite dans les colonnes qui leur sont attribuées. Ces codes I et U sont aussi numériques, pouvant varier chacun de 1 à 999 en ordre croissant de séquence: Ainsi, l'interview 002 ne peut précéder l'interview 001, et l'unité 005 ne peut précéder l'unité 002 à l'intérieur du même interview (--il y aurait un code d'erreur signalant une rupture de séquence--).

Quant au code S, il peut être représenté par tous les jeux de caractères décrits au point précédent, y compris les caractères de séparation de mots ou de phrases.

Les quatre lignes de codes de segmentation qui figurent dans le tableau ci-dessous sont lues de façon équivalente par les programmes CSCHECK et CSCUT: L'interview no.1 et l'unité no. 1 peuvent s'écrire (dans leurs zones respectives):

	7	8
	1234567890	
...	001	001A
...	1	1A
...	1	1 A
...	1	1A

L'utilisateur définit librement ce qu'il entend par interview, par unité, et par locuteur, ces trois modes de segmentation étant pures conventions. Il y a par contre quelque intérêt à faire

varier un mode plus vite qu'un autre. Par exemple, si les cinq chapitres d'un livre sont autant d'interviews, les paragraphes de chacun des chapitres en seront les unités. Cette façon de procéder permet ultérieurement de segmenter le corpus selon un mode ou selon l'autre ou encore selon une combinaison de ces modes. Pour poursuivre l'exemple commencé, si chaque chapitre comporte 50 paragraphes, on aura 5 interviews de 50 unités chacune; ceci peut donner lieu plus tard, soit à 5 segments --si on segmente par interview--, soit à 50 segments --si on segmente par unité--, soit encore à 250 segments --si on segmente en combinant unités et interviews--. D'autres possibilités de segmentation existent encore, qui font l'objet d'explications plus détaillées dans la section consacrée au programme CSCUT.

Les codes eux-mêmes peuvent être ajoutés au fichier-source soit via un éditeur de textes, soit via l'option FILL=1 du programme CSJOIN. Les modifications opérées par CSJOIN sont ensuite à modifier (éventuellement) via un éditeur de textes.

f. Exemple de fichier de données SOURCE.

L'extrait de fichier-source ci-dessous a été tronqué en sorte de pouvoir faire figurer sur cette page un fichier dont la longueur normale est 80 caractères par ligne. Les codes de segmentation qu'on trouve ci-dessous doivent donc, dans un fichier--source réel, se trouver en position 73 à 80.

*SOURCE-INTRUSE	
L'INTRUSE, DE J. L. BORGES, IN LE RAPPORT DE BRODIE	
*ENDCOM~	
ON DIT (MAIS C'EST PEU PROBABLE)	001 001A
QUE CETTE HISTOIRE FUT RACONTE1E PAR	001 001A
\$EDUARDO, LE CADET DES \$NILSEN, A2 LA	001 001A
VEILLE1E FUNE2BRE DE	001 001A
\$CRISTIAN, L'AI3NE1, QUI MOURUT DE MORT	001 001A
NATURELLE, VERS LES	001 001A
ANNE1ES 1890, DANS LA COMMUNE-N DE \$MORON.	001 001A
...(TEXTE INTERROMPU ICI)	
A2 \$TURDERA, ON LES APPELAIT LES \$NILSEN.	001 002A
LE CURE1 ME DIT QUE SON PRE1DE1CESSEUR SE	001 002A
SOUVENAIT D'AVOIR VU, NON	001 002A
SANS E1TONNEMENT, CHEZ CES GENS UNE VIEILLE	001 002A
\$BIBLE EN E1CRITURE	001 002A
GOTHIQUE, A2 RELIURE NOIRE; DANS LES DERNIE2RE	001 002A
PAGES IL AVAIT VU,	001 002A
INSCRITS A2 LA MAIN, DES NOMS ET DES DATES.	001 002A
C'E1TAIT LE SEUL LIVRE-N QU'IL Y EU3T DANS LA MAISON.	001 002A
LA DESTINE1E ITINE1RANTE DES \$NILSEN, PERDUE LA2	001 002A
COMME TOUT SE	001 002A
PERDRA.	001 002A
...(TEXTE INTERROMPU ICI)	
LE QUARTIER CRAIGNAIT CES ROUQUINS; IL N'E1TAIT PAS	001 002A
IMPOSSIBLE QU'ILS	001 002A
AIENT EU CERTAINS MEURTRES A2 LEUR ACTIF-N.	001 002A
ILS SE BATTIRENT UNE FOIS, CO3TE A2 CO3TE, CONTRE LA	001 002A
POLICE.	001 002A
ON DIT QUE LE CADET SE MESURA A2 \$JUAN \$IBERRA ET QU'	001 002A
IL N'EUT	001 002A
PAS LE DESSOUS, CE QUI, AU DIRE DES CONNAISSEURS,	001 002A
REPRE1SENTE UN	001 002A
EXPLOIT.	001 002A
...(TEXTE INTERROMPU ICI)	
CECI, ET LE RESTE-N QUE NOUS IGNORONS, PERMET DE	001 003A
COMPRENDRE LE BLOC	001 003A
QU'ILS FORMAIENT.	001 003A
SE FA3CHER AVEC L'UN, C'E1TAIT SE FAIRE DEUX ENNEMIS.	001 003A
LES \$NILSEN E1TAIENT COUREURS, MAIS LEURS AVENTURES	001 004A
AMOUREUSES	001 004A
AVAIENT E1TE1 JUSQU'ALORS DE CELLES QUI SE PASSENT	001 004A
SOUS UN PORTAIL OU	001 004A
DANS UNE MAISON CLOSE.	001 004A
LES COMMENTAIRES ALLE2RENT DONC BON TRAIN QUAND	001 004A
\$CRISTIAN AMENA	001 004A
CHEZ EUX \$JULIANA \$BURGOS.	001 004A
IL EST VRAI QU'IL Y GAGNAIT UNE SERVANTE, MAIS IL	001 004A
EST NON MOINS VRAI	001 004A

QU'IL LA COMBLAIT D'AFFREUX BIJOUX DE PACOTILLE ET	001	004A
QU'IL L'EXHIBAIT	001	004A
DANS LES BALS.	001	004A
DANS CES PAUVRES BALS DE QUARTIER, OÙ CERTAINES	001	004A
FIGURES-N DU TANGO	001	004A
ÉTAIENT INTERDITES ET OÙ L'ON DANSAIT ENCORE DANS	001	004A
DES SALLES BIEN	001	004A
ÉCLAIRÉES.	001	004A
\$JULIANA AVAIT LE TEINT MAT ET LES YEUX EN AMANDE; IL	001	004A
SUFFISAIT	001	004A
QU'ON LA REGARDAIT POUR QU'ELLE SOURÎT.	001	004A
DANS UN QUARTIER MODESTE, OÙ LE TRAVAIL ET LE MANQUE-N	001	004A
DE SOINS	001	004A
ABÎMENT LES FEMMES, ELLE PASSAIT POUR JOLIE.	001	004A
AU DÉBUT, \$EDUARDO LES ACCOMPAGNAIT.	001	005A
PUIS IL DUT SE RENDRE À \$ARRECIFES POUR JE NE SAIS	001	005A
QUELLE	001	005A
AFFAIRE; À SON RETOUR IL AMENA À LA MAISON UNE JEUNE	001	005A
FEMME QU'IL	001	005A
AVAIT TROUVÉE SUR SA ROUTE SUR SA ROUTE ET QU'IL	001	005A
RENVOYA AU BOUT DE	001	005A
QUELQUES JOURS.	001	005A
IL SE RENFROGA; IL S'ENIVRAIT SEUL AU BISTROT ET NE	001	005A
PARLAIT À PERSONNE.	001	005A
IL ÉTAIT AMOUREUX DE LA FEMME DE \$CRISTIAN.	001	005A
LE QUARTIER, QUI S'EN APERCUT PROBABLEMENT AVANT LUI,	001	005A
PRÉVIT AVEC	001	005A
UNE JOIE PERFIDE LA RIVALITÉ QUI ALLAIT S'ENSUIVRE	001	005A
ENTRE LES DEUX	001	005A
FRÈRES.	001	005A
UN SOIR QU'IL RENTRAIT TARDIVEMENT DU BISTROT DU COIN,	001	006A
\$EDUARDO	001	006A
VIT LE CHEVAL NOIR DE \$CRISTIAN ATTACHÉ À LA	001	006A
PALISSADE.	001	006A
DANS LA COUR, L'ÂNE L'ATTENDAIT DANS SES PLUS BEAUX	001	006A
HABITS.	001	006A
LA FEMME ALLAIT ET VENAIT, UN POT DE MATEL À LA MAIN.	001	006A
\$CRISTIAN DIT À \$EDUARDO:	001	006A
JE M'EN VAIS À UNE FÊTE CHEZ \$FARIAS.	001	006B
JE TE LAISSE \$JULIANA; SI TU LA VEUX, TU PEUX LA PRENDRE	001	006B
C'ÉTAIT DIT D'UN TON-N À LA FOIS AUTORITAIRE ET	001	007A
CORDIAL. \$EDUARDO LE REGARDA LONGUEMENT; IL NE SAVAIT	001	007A
QUE FAIRE.	001	007A
\$CRISTIAN SE LEVA, PRIT CONGEL D'\$EDUARDO, NÉGLIGEANT	001	007A
\$JULIANA QUI N'ÉTAIT POUR LUI QU'UN OBJET, MONTA À	001	007A
CHEVAL ET PARTIT AU PETIT TROT, SANS SE PRESSER.	001	007A
À DATER DE CETTE NUIT-N LÀ, ILS SE LA PARTAGÈRENT.	001	008A
PERSONNE NE CONNAÎTRA LES DÉTAILS	001	008A
DE CE SORDIDE MÉNAGE-N À TROIS,	001	008A
QUI SCANDALISAIT LE QUARTIER.	001	008A

F. Les fichiers de données INROOT

Un fichier de type INROOT s'annonce, comme un fichier de type SOURCE, par trois types d'enregistrements:

1. Le premier enregistrement est constitué de:

```
*INROOT-iiiiiiiiccccccc.....c
```

a. où iiiiii est à remplacer par un nom de 1 à 8 caractères alignés à gauche, permettant d'identifier le fichier au cours des utilisations qui en seront faites plus tard. Peuvent entrer dans la composition de ce nom, les lettres A à Z, les chiffres 0 à 9, les caractères (entre < et >, non compris) <+ & \$ * % _ @ -> ,

b. où ccc...c est un commentaire facultatif occupant la zone 17 à 80.

2. Le deuxième type d'enregistrement, facultatif, est constitué de commentaires occupant la zone 1 à 80 sur un maximum de 999 lignes. Trouvent place ici des observations et mémos sur les conditions où le fichier INROOT a été créé, son ou ses auteurs, la date de sa création, ainsi de suite.

3. Le troisième type d'enregistrement, obligatoire et unique, est constitué de:

```
*ENDCOM~
```

débutant en colonne 1, et qui annonce la fin de la zone de commentaires, même en l'absence de ceux-ci.

4. Format d'un fichier de données INROOT.

Les trois enregistrements précédents sont suivis des données INROOT proprement dites. Un fichier de type INROOT a pour but principal de ramener les formes différentes des mots (féminin, pluriel, et formes conjuguées) à une forme unique minimale ou à un radical. On peut donc s'attendre à trouver dans un fichier INROOT des radicaux de mots d'une part, et d'autre part toutes les formes différentes des mots, transformables en une base unique et univoque. Notons qu'il est aussi permis d'utiliser un fichier de type INROOT pour transformer les séparateurs, marqueurs, et autres signes de ponctuation.

En l'occurrence, les formes différentes des mots prennent place, à raison d'une forme par ligne, dans les colonnes 1 à 20, tandis que les bases prennent place dans les colonnes 24 à 43. Les colonnes 21 à 23 et 44 à 46 restent vides, les colonnes 47 à 80 pouvant accueillir un commentaire.

Note:

On peut élaborer un nouveau fichier INROOT, ou en modifier un ancien, de façon manuelle. Mais on peut aussi utiliser le programme CWKWOC (avec l'option "PUNCH = 2") pour générer de nouvelles entrées à inclure dans un fichier INROOT. On se reportera au programme CWKWOC pour les détails concernant la génération automatique de ce type de fichier.

5. Lexies à inclure et à exclure dans un fichier INROOT.

Un fichier de données INROOT peut contenir quatre types de lexies. On peut d'abord transformer un mot en un (autre) mot, soit "HEUREUSEMENT" en "HEUREUX". On peut ensuite transformer une racine (ou radical) en une autre racine, soit "NPP." en "\$..".

(Dans le système PROTAN, une racine de mot se caractérise par la présence d'un point <.> comme dernier caractère de la chaîne de caractère, par exemple "HEUREU."). On peut encore transformer une racine¹⁰ en un mot, soit "HEUR." en "HEUREUX". Et on peut enfin transformer une racine ou un mot en rien, la zone correspondante dans la colonne de droite restant alors vide, soit les mots "DU", "DE", "LA", et "LE" à gauche (colonnes 1 à 20) et rien à droite¹¹ (colonnes 24 à 43).

Mots et racines peuvent être suivis de commentaires (jusqu'à la colonne 20 pour la zone de gauche, et la colonne 43 pour la zone de droite). Pour les mots, le commentaire doit être précédé d'un blanc <~> au moins et d'une parenthèse ouverte <~(>. Pour les racines, le commentaire doit être précédé d'une parenthèse ouverte, précédée ou non de blancs <~> après le point qui caractérise une racine <~(>.

Lorsqu'un fichier de données INROOT, transformé par CRCHECK en un fichier-système "ROOTS", est appliqué à un corpus, il opère alors une série de réductions sur le texte. Ces réductions sont fonction du type de lexies contenues dans le fichier de données INROOT. Dans le premier cas (mot en mot), toutes les occurrences de "HEUREUSEMENT" sont transformées en "HEUREUX". Dans le deuxième cas (racine en racine), tous les mots commençant par les caractères "NPP" sont transformés en mots commençant par le caractère "\$". Dans le troisième cas (racine en mot), tous les mots commençant par "heur" (c'est-à-dire HEUREUSEMENT, HEUREUX, HEUREUSE, et HEUREUSES) sont transformés en un seul type "HEUREUX". Dans le quatrième cas (mot ou racine en rien), toutes les occurrences de "DU", "DE", "LA", et "LE" sont purement effacées du corpus. Si on a rassemblé tous les mots-outils de la langue (articles définis et indéfinis, pronoms, prépositions, etc.) dans un fichier "INROOT" séparé, on peut, à un certain moment, décider d'effacer du texte tous ces mots-outils qui l'encombrent. Ces mots ne sont pas très nombreux, mais épuisent une grande partie de la fréquence totale des mots. On s'en aperçoit en exécutant le programme CWTALLY sur un corpus.

Par ailleurs, il se peut qu'on veuille utiliser la transformation "racine en mot" (par exemple "HEUR." en "HEUREUX") tout en devant procéder à certaines exclusions, par exemple ne pas inclure la forme "HEURE" dans la transformation de "HEUR." en "HEUREUX". Pour obtenir ce résultat, il suffit de mettre la même forme dans les deux colonnes de données du fichier, par exemple "HEURE" en colonnes 1 à 20 et "HEURE" en colonnes 24 à 43, ce qui signifie "maintenir inchangé" ou, si on veut, "à exclure de toute transformation.

6. Exemple de fichier de données INROOT.

¹⁰ La transformation d'un mot en une racine n'est pas autorisée et n'aurait d'ailleurs pas beaucoup de sens.

¹¹ Sauf commentaire éventuel, commençant au plus tôt en colonne 25 par une parenthèse ouverte.

```

*INROOT-FRSTRP01   SOURCE: HOGENRAAD, MONOLOGUE
*ENDCOM~
$AME1RICA1NS       $AME1RICA1N
$BELGES             $BELGE
$FRANC1AISES        $FRANC1AIS
$MASSAI4S           $MASSAI4
A                   AVOIR
A-EU-MARRE          AVOIR-MARRE
A-MARRE             AVOIR-MARRE
A-T-ON              AVOIR
ABAISSSE            ABAISSER
ABAISSSENT          ABAISSER
ABANDONNE           ABANDONNER
ABBE1S              ABBE1
ABEILLES            ABEILLE
ABHORRE             ABHORRER
ABI3MENT            ABI3MER
ABOIE               ABOYER
ABOMINABLEMENT      ABOMINABLE
ABOUTIT            ABOUTIR
ABOYAIT             ABOYER

```

G. Les fichiers de données INDICT

Les formats et paramètres des fichiers de données INDICT varient quelque peu selon qu'on doit créer un dictionnaire de normes ou un dictionnaire de catégories. D'où la distinction qui suit.

1. Les fichiers INDICT dans les dictionnaires de normes.

Ces fichiers s'annoncent par quatre types d'enregistrements:

a. Le premier enregistrement est constitué de:

```
*INDICN-iiiiiiiccccc.....c
```

(1) où iiiiiiii est à remplacer par un nom de 1 à 8 caractères alignés à gauche, permettant d'identifier le dictionnaire. Peuvent entrer dans la composition de ce nom, les lettres A à Z, les chiffres 0 à 9, les caractères (entre < et >, non compris) <+ & \$ * % _ @ ->.

(2) où ccccc...c est un commentaire facultatif occupant la zone 17 à 80.

b. Le deuxième type d'enregistrement, facultatif, est constitué de commentaires occupant la zone 1 à 80 sur un maximum de 999 lignes. Trouvent place ici des observations et mémos sur les conditions où le fichier INDICT a été créé, son ou ses auteurs, la date de sa création, ainsi de suite.

c. Le troisième type d'enregistrement, obligatoire et unique, est constitué de:

```
*ENDCOM~
```

débutant en colonne 1, et qui annonce la fin de la zone de commentaires, même en l'absence de ceux-ci.

d. Le quatrième type d'enregistrement est constitué d'une ligne par sous-dictionnaire, chaque ligne étant constituée

(1) de la borne inférieure du sous-dictionnaire (colonnes 1 à 3),

(2) de la borne supérieure du sous-dictionnaire (colonnes 5 à 7),

(3) de son identification (colonnes 9 à 16),

(4) et d'un commentaire libre (servant aussi de titre de page) (colonnes 17 à 80).

2. Les fichiers INDICT dans les dictionnaires de catégories.

Quatre types d'enregistrements annoncent ces fichiers:

a. Le premier enregistrement est constitué de:

```
*INDICC-iiiiiiiiccccccc...c
```

(1) où iiiiii est à remplacer par un nom de 1 à 8 caractères alignés à gauche, permettant d'identifier le dictionnaire. Peuvent entrer dans la composition de ce nom, les lettres A à Z, les chiffres 0 à 9, les caractères (entre < et >, non compris) <+ & \$ * % _ @ ->.

(2) où ccccc...c est un commentaire facultatif occupant la zone 17 à 80.

b. Le deuxième type d'enregistrement, facultatif, est constitué de commentaires occupant la zone 1 à 80 sur un maximum de 999 lignes. Trouvent place ici des observations et mémos sur les conditions où le fichier INDICT a été créé, son ou ses auteurs, la date de sa création, ainsi de suite.

c. Le troisième type d'enregistrement, obligatoire et unique, est constitué de:

```
*ENDCOM~
```

débutant en colonne 1, et qui annonce la fin de la zone de commentaires, même en l'absence de ceux-ci.

d. Le quatrième type d'enregistrement, facultatif, est constitué, pour ces dictionnaires de catégories, en colonnes 1 à 3, du numéro de la catégorie, et en colonnes 10 à 39, de l'intitulé de la catégorie. Cette séquence est répétée autant de fois qu'il y a de catégories.

3. Lexies à inclure et à exclure

Comme pour les fichiers de données INROOT, on peut inclure ou exclure, d'un dictionnaire de normes ou d'un dictionnaire de catégories, des mots ou racines de mots, celles-ci se terminant par un point comme dernier caractère.

Pour les deux types de dictionnaires alors, viennent, après les enregistrements 1 à 4 qui leur sont spécifiques, les mots et racines de mots à inclure ou à exclure. Mots et racines se placent en colonnes 4 à 23, alignés à gauche et suivis, en format libre, du ou des numéros de catégories à affecter au mot ou à la racine, par groupes de 3 colonnes à partir de la colonne 24 incluse. Un maximum de 10 affectations de catégories est autorisé par mot; les mots et racines des dictionnaires de normes sont affectés d'autant de valeurs (par tranches de 3 colonnes) qu'il y a de sous-dictionnaires.

Mots et racines peuvent être suivis de commentaires (jusqu'à la colonne 23 incluse). Pour les mots, le commentaire doit être précédé d'un blanc (~) au moins et d'une parenthèse ouverte (<(>). Pour les racines, le commentaire doit être précédé d'une parenthèse ouverte (précédée ou non de blancs (~) après le point qui caractérise une racine.

On inclut un mot ou une racine en lui affectant un numéro de catégorie, comme ci-dessous, et on l'exclut en ignorant la zone d'affectation de catégorie qui reste alors vide.

4. Exemples de fichiers de données INDICT

a. Le dictionnaire de normes d'imagerie DIMAB.

```
*INDICN-DIMAB    DICTIONNAIRE D'IMAGERIE (R. HOGENRAAD)
*ENDCOM~
10 70 DIMAB    DICTIONNAIRE D'IMAGERIE (R. HOGENRAAD)
  ATMOSPHE2RE.(S      35
  ATTACHE          36
  ATTACHE-N.       36
  ATTACHES         36
  ATTACHES-N.      36
  ATTENTION        30
  ATTENTIONS       30
  ATTITUDE.(S      27
  AUDITOIRE.(S     49
  AUTEUR.(S        28
  AVION            60
  BARRIE2RE.(S     58
  BATEAU.(X        61
  BA3TIMENT.(S     54
  BEFFROI.(S       54
  BESOIN.(S        27
  BIBLIOTHE2QUE.(S  56
  BILAME.(S        32
  BLAGUE           35
  BLAGUE-N.        35
```

b. Le dictionnaire de catégories d'imagerie régressive DIRE

*INDICC-DIRE IMAGERIE REGRESSIVE (R. HOGENRAAD)

*ENDCOM~

1	ORALITE
2	ANALITE
3	SEXE
4	SENSATION GENERALE
5	TOUCHER
6	GOUT
7	ODORAT
8	OUIE
9	VUE
10	FROID
11	DUR
12	DOUX
13	PASSIVITE
14	VOYAGE
15	MOUVEMENT NON ORIENTE
16	DIFFUS
17	CHAOS
18	INCONNU
19	INTEMPOREL
20	ALTERATION DE LA CONSCIENCE
21	FRANCHISSEMENT & PASSAGE
22	NARCISSISME
23	CONCRET
24	MONTER
25	HAUT
26	DESCENDRE
27	PROFONDEUR
28	FEU
29	EAU
30	AFFECT POSITIF
31	ANXIETE
32	TRISTESSE
33	AMOUR
34	AGRESSION
35	COMPORTEMENT EXPRESSIF
36	TRIOMPHE
37	PENSEE ABSTRAITE
38	COMPORTEMENT SOCIAL
39	COMPORTEMENT INSTRUMENTAL
40	LOI & RESTRICTION
41	ORDRE
42	REFERENCE TEMPORELLE
43	IMPERATIF MORAL
44	BESOIN :.....01-03
45	SENSATION :.....04-12
46	SYMBOLISATION DEFENS :..13-17
47	CONNAISSANCE REGRESS :..18-23
48	IMAGERIE ICARIENNE :....24-29
49	PROCESSUS PRIMAIRES :...01-29
50	EMOTIONS :.....30-36
51	PROCESSUS SECONDAIRES :.37-43

ABATTEMENT.(S	32	50
ABHOR.(ER	34	50
ABI3ME.(S	27	48 49
ABOIE.(MENT	35	50
ABOYER	35	50
ABRIT.(ER	38	51
ABRUPT.(ES	25	48 49
ABSCONS.(ES	37	51
ABSINTHE.(S	1	44 49
ABSTRACTION.(S	37	51
ABSTRUS.(ES	37	51
ABUS.(ER	34	50
ABYSSA.(UX	27	48 49
ACARIA3TRE.(S	34	50
ACCEPT.(ATION	38	51
ACCE1LE1RATION.(S	42	51
ACCE1LE2RE	42	51
ACCE2S	21	47 49

5. Les fichiers de données INFIND

Les fichiers INFIND débutent par quatre types d'enregistrement, soit:

a. Le premier enregistrement, constitué de:

```
*INFIND-iiiiiiiiccccccc.....c
```

(1) où iiiiiiic est à remplacer par un nom de 1 à 8 caractères alignés à gauche, permettant d'identifier le fichier au cours des utilisations qui en seront faites plus tard. Peuvent entrer dans la composition de ce nom, les lettres A à Z, les chiffres 0 à 9, les caractères (entre < et >, non compris) <+ & \$ * % _ @ -> ,

(2) où ccc...c est un commentaire facultatif occupant la zone 17 à 80.

b. Le deuxième type d'enregistrement, facultatif, est réservé aux commentaires. Ils peuvent s'étendre sur la zone 1 à 80 et contenir jusqu'à 999 lignes. Trouvent place ici des observations et mémos sur les conditions où le fichier INFIND a été créé, son ou ses auteurs, la date de sa création, ainsi de suite. S'agissant d'un fichier de recherche d'adresses de mots souvent considérés comme ambigus par l'analyste, on mettra dans ces commentaires, par exemple, les raisons pour lesquelles un tel fichier de mots a été constitué.

c. Le troisième type d'enregistrement, obligatoire et unique, est constitué de:

```
ENDCOM~
```

débutant en colonne 1; cet enregistrement sert à délimiter la fin des commentaires, même en l'absence de ceux-ci, et le début du fichier de travail proprement dit.

d. Format d'un fichier de données INFIND.

Les trois enregistrements précédents sont suivis des données INFIND qui constituent le quatrième type d'enregistrement. Les mots ou racines de mots y occupent les positions 1 à 20, la position 21 ne pouvant éventuellement être occupée que par un tiret (-) pour signifier l'exclusion du mot ou de la racine de l'opération de recherche d'adresse. Les colonnes 24 à 80 sont réservées à des commentaires qui ne sont mémorisés que s'ils figurent dans la zone 24-43.

e. Exemple de fichier de données INFIND

```
*INFIND-ISIDA
*ENDCOM~
BASE
CAUSE
CLASSE
CONSERVE
DONNE1E
DROIT
ENTREPRISE
LAVE
MARCHE1
MASSE
MISE
PORTE
PORTE1E
PRISE
RE1SERVE
RE2GLE
SOMMES
SOUFFLE
TABLE
RE2VE
VERS
CHER
CONTINENT
COURS
DESTINE1E
DE1COUVERTE
E1CONOMIES
E1TONNANT
INDE1PENDANT
JUGE
GRAVE
MANUFACTURE
COMPTE
ENTRE
MANQUE
MET
PART
PARTI
PARTIE
PASSANT
RESTE
PASSE1
PERMIS
POUVOIR
PRODUIT
PUIS
RECHERCHE
```

6. Note sur la syntaxe des fichiers de paramètres et options.

Les fichiers de paramètres et options s'écrivent selon une syntaxe simple où chaque option, avec sa valeur (signe "=" entre option et valeur), est séparée de la suivante par une virgule et/ou un ou plusieurs blancs (~), la dernière option étant toujours suivie d'un point-virgule. L'ordre de présentation des options n'a pas d'importance par ailleurs. On aura donc, par exemple (entre < et > non compris): <PRINTT = 1, CONT = 1, PUNCH = 2;> qui sont des options possibles pour le programme CDWLOOK.

La syntaxe des paramètres est, elle, propre à chaque programme où elle est détaillée conformément.

7. Note générale sur les majuscules et minuscules dans PROTAN.

Le système PROTAN ne reconnaît pas la différence entre les majuscules et les minuscules; ces dernières, autorisées, sont cependant converties en majuscules lors du traitement par n'importe quel programme. La seule exception à cette règle est l'emploi différencié des majuscules et minuscules dans la zone de "speaker" (colonne 80) des fichiers SOURCE.

V. Les programmes du système PROTAN

Schéma de présentation des programmes

Chacun des programmes analysés dans ce chapitre est présenté selon un ordre standard qui facilite le repérage des informations spécifiques qu'on cherche. Pour chaque programme, on trouvera, dans l'ordre, les rubriques suivantes:

1. But
2. Exécution
 - a. Entrées/Sorties
 - b. Paramètres et options
3. Exemples
4. Préparation des données
5. Autres informations
6. Remarques
7. Messages du programme

1. But

Ce que fait le programme et à quoi servent les éventuels fichiers-système qu'il crée. La place de ce programme dans la logique du système.

2. Exécution

Tout ce dont il est nécessaire de disposer pour que le programme soit exécuté, c'est-à-dire:

a. Entrées/Sorties: Les fichiers qui doivent nécessairement être présents pour que le programme soit exécuté, et les fichiers qu'il crée éventuellement, voire par défaut.

b. Paramètres et options: Ce que fait et crée le programme peut varier selon la présence ou l'absence de paramètres et options contraignants s'ils ont été prévus. Ces paramètres et options constituent l'essentiel des informations que l'utilisateur doit fournir au programme de traitement.

Sauf erreur, PROTAN produit toujours une liste, même en l'absence d'un fichier de paramètres --PROTAN cherche dans ce cas le nom par défaut du fichier de paramètres et options.

3. Exemples

Lorsque l'exécution ou les résultats d'un programme sont complexes, des exemples d'exécutions différentes seront présentés en même temps que les résultats produits par cette exécution. Les résultats sont d'ailleurs souvent plus complexes que la commande d'exécution elle-même.

4. Préparation des données

Les fichiers que les programmes utilisent en entrée doivent toujours se présenter selon un certain format, variable d'un programme à l'autre, qui doit être respecté au risque de non-exécution du programme.

5. Autres informations

On trouvera ici toutes les informations, souvent non techniques, sur le sens du programme, son utilité dans l'analyse, et, de façon générale, toute information susceptible de permettre une utilisation intelligente du système.

6. Remarques

Souvent, des remarques d'ordre technique permettent à l'utilisateur débutant de se servir du système sans aucune erreur.

7. Messages du programme

Chaque programme signale à l'utilisateur, par des messages spécifiques, des informations, des avertissements ou des erreurs dont certaines n'entraînent même pas nécessairement l'interruption du programme. Chaque programme signale également s'il a fonctionné sans erreur jusqu'à la fin.

VI. Le programme CDCHECK

A. But du programme

Le but de CDCHECK est de créer le fichier-système de type "DICTS" qui contient les mots du dictionnaire de catégories qui sera utilisé par le programme CDWLOOK pour comparer les mots du texte aux catégories du dictionnaire. (Voir Chapitre XXXIII. "Annexes: Les dictionnaires de formes nominales et de catégories").

B. Exécution

1. Entrées/sorties: Comme tous les programmes de PROTAN, CDCHECK demande un fichier de paramètres et options en entrée et crée un fichier listing en sortie. Le fichier de paramètres et options doit identifier les entrées et sorties, soit:

a. **Entrées:** un ou des fichiers INDICT (partiels) contenant les mots du dictionnaire "à l'état brut". Ces noms de fichiers sont donnés via l'option SINDICT (voir paramètres et options).

b. **Sortie(s):** En sortie, le fichier de paramètres et options identifie le nom du fichier-système de type "DICTS" pour les programmes CDLISTA, CDLISTC, CDWJUXT, et CDWLOOK. Le nom de ce fichier-système est donné par l'option "SDICTS = ".

2. Paramètres et options;

SINDICT = '*' ; cette instruction renvoie à une liste de (chemin et) noms de fichiers INDICT après les options (à partir d'une autre ligne et sans apostrophes).
= 'chemin et nom de fichier' ; cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type INDICT.
Option pour DOS et UNIX.

SDICTS = 'chemin et nom du fichier devant contenir le dictionnaire à créer' (fichier "DICTS").
Option pour DOS et UNIX.

LIST1 = Par défaut ("LIST1 = 0"), il n'y a pas impression de la phase 1 de l'exécution de CDCHECK. Autrement ("LIST1 = 1"), la phase 1 est imprimée.
La phase 1 de l'exécution d'un dictionnaire est la phase où celui-ci apparaît tel qu'il a été construit, les mots et racines à inclure et à exclure y étant rangés dans l'ordre original.

Cette première phase est suivie d'une phase 2 où les mots du dictionnaire sont rangés dans un ordre alphabétique d'où sont effacés les mots à propos desquels une erreur de syntaxe a été détectée lors de la phase 1.

LIST2 = Par défaut ("LIST2 = 0"), il n'y a pas impression de la phase 2. "LIST2 = 1" permet l'impression de la phase 2.

DICT = "DICT = 1", le défaut, entraîne la création d'un fichier-système de type "DICTS", utilisé par CDLISTA, CDLISTC, CDWJUXT, et CDWLOOK. "DICT = 0" inhibe la création du fichier de type "DICTS".

STEMP = 'c:...*.tmp' (défaut, *STEMP* = '*.tmp'). Voir le Chapitre III. Description (D.3. Remarques générales concernant les applications de PROTAN en DOS et UNIX).

C. Exemples

1. Exemple de fichier d'exécution

Contenu du fichier "cdcheck1.bat":

```
call protan.bat cdcheck diredch.lis diredch.dch
```

2. Exemple de fichier de paramètres et options

Contenu du fichier de paramètres et options "diredch.dch

```
SINDICT = '*', SDICTS = 'c:\protan\data.ind\ffrid\ffrid.dts';  
c:\protan\data.ind\ffrid\dire0.ind  
c:\protan\data.ind\ffrid\dire1.ind  
c:\protan\data.ind\ffrid\dire2.ind  
c:\protan\data.ind\ffrid\dire3.ind
```

Toutes les options LIST1, LIST2, et DICT sont prises par défaut et un fichier-système "ffrid.dts" sera créé.

D. Préparation des données

Aucune.

E. Autres informations

Aucune.

F. Remarques

Aucune.

G. Messages propres au programme CDCHECK

*CDCH201E	duplicate words/roots = nnnnnn
*CDCH202E	invalid category definition
*CDCH203E	category 0 or > 255 not available
*CDCH204E	duplicate category definition
*CDCH205W	label left truncated
*CDCH206E	syntactical conflict
*CDCH207I	no valid user category defined
*CDCH208U	user's dictionary length > 99999
*CDCH209E	inaccurate word/root alignment
*CDCH210E	null word/root
*CDCH211S	no word/root to include
*CDCH214E	word/root syntax error
*CDCH215E	sub-dictionary nn : tag out of range
*CDCH217S	heterogeneous tag list
*CDCH219E	invalid category assignment
*CDCH220W	category nnn undefined by user
*CDCH221E	duplicate category assignment
*CDCH222E	significant char. after tag end
*CDCH223E	words/roots purged = nnnnnn
*CDCH224W	category nnn is never referenced
*CDCH225E	duplicate root - latest one in use
*CDCH226E	duplicate word - first one in use
*CDCH227S	no sub-dictionary defined
*CDCH228S	invalid lower boundary
*CDCH229S	lower boundary 0 or > 255 not available
*CDCH230S	inaccurate lower/upper boundary alignment
*CDCH231S	invalid upper boundary
*CDCH232S	upper boundary 0 or > 255 not available
*CDCH233S	more than 10 sub-dictionaries
*CDCH234S	duplicate sub-dictionary identification

VII. Le programme CDLISTA

I. But du programme

Ce programme permet l'impression des dictionnaires de catégories ou de normes en ordre alphabétique sur les mots.

A. Exécution

1. Entrées/sorties: fichier de paramètres et options en entrée, fichier listing en sortie. Le fichier de paramètres et options contient les noms des entrées et sorties suivantes:

a. **Entrée:** le nom du fichier de type DICTS, c'est-à-dire le nom du dictionnaire (par l'option SDICTS=).

b. **Sortie:** néant.

2. Paramètres et options:

SDICTS = pour fournir au programme le nom (et le chemin éventuellement) du fichier DICTS. Par exemple, *SDICTS* = 'd:....UCLdch.dts'. Option pour DOS et UNIX.

B. Exemples

1. Exemple de fichier d'exécution

Contenu du fichier "cdlista1.bat":

```
call protan.bat cdlista diredla.lis diredla.dla
```

2. Exemple de fichier de paramètres et options

Contenu du fichier de paramètres et options "diredla.dla":

```
SDICTS = 'd:\protan\data.ind\dire\diredch.dts';
```

C. Préparation des données

Aucune.

D. Autres informations

Aucune.

E. Remarques

Aucune.

F. Messages propres au programme:

```
*CDLA241E duplicate root - latest one in use
*CDLA242E duplicate word - first one in us
```


VIII. Le programme CDLISTC

A. But du programme

Ce programme permet l'impression des dictionnaires de catégories ou de normes par catégorie sur les codes de catégories ou de normes. Dans le cas d'un dictionnaire de normes, il y a lieu de coder le sous-dictionnaire qu'on désire imprimer.

B. Exécution

1. Entrées/sorties: fichier de paramètres et options en entrée, fichier listing en sortie. Le fichier de paramètres et options contient les noms des entrées et sorties suivantes:

a. **Entrée:** le nom du fichier de type DICTS, c'est-à-dire le nom du dictionnaire (par l'option "SDICTS=").

b. **Sortie:** néant.

2. Paramètres et options:

SDICTS = Cette option fournit au programme le nom (et éventuellement le chemin) du fichier DICTS. Par exemple, *SDICTS* = 'd:....UCLdch.dts'. Option pour DOS et UNIX.

SUBDICT= Nécessaire pour identifier un sous-dictionnaire d'un dictionnaire de normes au cas où il y en a plus d'un.

C. Exemples

1. Exemple de fichier d'exécution

Contenu du fichier "cdlistc1.bat":

```
call protan.bat cdlistc ucldlc.lis ucldlc.dlc
```

2. Exemple de fichier de paramètres et options

Contenu du fichier "ucldlc.dlc":

```
sdicts = 'd:\protan\tests\data.ind\ucl\uclch.dts',  
subdict = 'uclem';
```

On utilise le sous-dictionnaire d'émotionnalité du dictionnaire de normes UCL.

D. Préparation des données

Aucune.

E. Autres informations

Aucune.

F. Remarques

Aucune.

G. Messages propres au programme

*CDLC261S sub-dictionary "xxxxxxx" not found
--

IX. Le programme CDWJUXT

A. But du programme

Le but du programme CDWJUXT est d'identifier dans un texte toutes les phrases où on trouve des mots appartenant à deux dictionnaires différents ou à deux catégories différentes d'un même dictionnaire.

On cherche donc des co-occurrences, ou juxtapositions, qui seront converties par après en un rapport quelconque, par exemple, le nombre de co-occurrences divisé par le nombre de mots dans le segment.

Comme on ne cherche que des co-occurrences, on peut faire interagir des dictionnaires de structures différentes, c'est-à-dire, chercher, par exemple, les juxtapositions d'un dictionnaire de catégories et d'un autre dictionnaire de normes. On peut aussi chercher les occurrences de phrases où se trouvent des mots ayant des valeurs extrêmes sur la même dimension. Par exemple, si le contraste est une condition de la métaphore, on pourrait chercher à connaître toutes les phrases où il y a à la fois des mots ayant de très hautes valeurs d'imagerie (des valeurs entre 50 et 70) et des mots ayant des valeurs très basses d'imagerie (entre 10 et 30).

Par exemple, Bucci (1985) avance l'idée que les moments de transfert réussi, en psychanalyse, seraient caractérisés par les moments de la thérapie où on trouve à la fois des images fortes et des émotions intenses. (Voir aussi Thomä et Kächele, 1992, en particulier la section intitulée "Theoretical remarks about a 'good hour', pp. 471-480). On pourrait vérifier cette hypothèse en cherchant, dans des protocoles de thérapie, toutes les phrases où se retrouvent des mots très chargés d'émotion (dictionnaire UCL, sous-dictionnaire d'émotionnalité) et très chargés en imagerie (dictionnaire DIMAB, ou dictionnaire de Desrochers et Bergeron, 1992).

B. Exécution

1. Entrées/sorties: Comme tous les programmes de PROTAN, CDWJUXT demande toujours un fichier de paramètres et options en entrée et créera toujours un fichier listing en sortie. Le fichier de paramètres et options doit identifier les **entrées** et **sorties**, soit:

a. Entrées:

(1) le nom d'un fichier WORDS, via l'option "SWORDS=",
(2) le nom d'un dictionnaire via l'option "SDICTS1=",
(3) et éventuellement, le nom d'un second dictionnaire, via l'option "SDICTS2=". Si on oppose des catégories d'un même dictionnaire, "SDICTS2 =" ne doit pas être précisé. On considère en effet que SDICTS2 = SDICTS1.

b. **Sortie:** En sortie, le fichier de paramètres et options identifie le nom du fichier PUNCHJ créé au moyen de l'option "SPUNCH=". Ce fichier PUNCH contient les résultats perforés prêts pour analyses statistiques ultérieures.

2. Paramètres et options:

-
- STAT* = Par défaut ("STAT = 0"), il n'y a pas impression des statistiques de tri qui accompagnent les fichiers de type "WORDS". "STAT = 1" déclenche l'impression de ces statistiques, c'est-à-dire, par segment et pour le total, le nombre de mots différents et le nombre total de mots.
- CONTEXT* = On peut se contenter d'un score global de juxtapositions, c'est-à-dire, de la proportion de co-occurrences de mots chargés en deux dimensions. On utilisera alors l'option "CONTEXT = 0".
Mais si on souhaite voir où et comment ces co-occurrences se passent, et quels sont les mots qui sont responsables de ces co-occurrences, alors, on doit utiliser l'option "CONTEXT = 1" qui imprime les phrases où de telles co-occurrences ont lieu.
A noter que les comparaisons entre dictionnaires(s) et corpus se font sur la génération la plus récente du fichier WORDS strippé; par contre, l'impression via CONTEXT se fait en utilisant le niveau 0 du fichier WORDS, c'est-à-dire, la sortie non strippée de CSCUT.
- LL* = L'option LL gère la longueur de ligne du texte, c'est-à-dire le nombre de caractères par ligne imprimée. Par défaut, LL est égal à 80; on peut le faire varier de 36 à 132.
- SDICTS1* = fournit au programme le nom (et le chemin éventuellement) du premier fichier DICTS. Par exemple,
SDICTS1 = 'd:....UCLdch.dts'.
Option pour DOS et UNIX.
- SDICTS2* = fournit au programme le nom (et le chemin éventuellement) du second fichier DICTS. Par exemple,
SDICTS2 = 'd:....UCLdch.dts'.
Option pour DOS et UNIX.
- SUBDICT1* = identifie un sous-dictionnaire du premier dictionnaire (de normes). Cette option est inutile s'il n'y a qu'un seul sous-dictionnaire. Un nom de sous-dictionnaire est par exemple "SUBDICT1 = 'uclem'", le sous-dictionnaire d'émotionnalité du dictionnaire UCL de normes d'émotions.
- SUBDICT2* = identifie un sous-dictionnaire du second dictionnaire (de normes). Cette option est inutile s'il n'y a qu'un seul sous-dictionnaire.
- SWORDS* = fournit au programme le nom et le chemin du fichier WORDS. Par exemple,

SWORDS = 'd:....rifrrws.wds'.
Option pour DOS et UNIX.

SPUNCH = fournit au programme le nom et le chemin
 du fichier de type PUNCHJ. Par exemple,
SPUNCH = 'd:....rifrimem.pcj'. Option pour
DOS et UNIX.

STEMP = fournit un nom générique de fichiers de
 travail temporaires. Ces fichiers sont dé-
 truits à la fin du travail par les procé-
 dures *protan.bat* (en DOS) ou *protan* (en
 UNIX). Pour plus de détails, se référer au
 Chapitre III. Description (D.3. Remarques
 générale concernant les applications de
 PROTAN en DOS et en UNIX).

PUNCH = "PUNCH = 0", le défaut, ne donne lieu à
 aucune perforation de résultats. "PUNCH =
 1" perfore les résultats imprimés sous la
 forme d'un fichier de type "PUNCHJ".

Il faut ajouter, sur une ligne suivante après le point-
virgule qui termine les options, les indications qui permettent au
programme CDWJUXT de savoir quelles sont les catégories pour
lesquelles il faut rechercher une co-occurrence par phrase, ou
quelles sont, pour les dictionnaires de normes, les bornes par
lesquelles on veut limiter la recherche de co-occurrences. On aura
donc par exemple:

41-70 / 41-70 ... pour indiquer qu'on souhaite
 opposer les valeurs élevées
 d'un dictionnaire de normes
 aux valeurs élevées d'un autre
 dictionnaire de normes;

41-70 / 10-40 ... pour indiquer qu'on souhaite
 opposer les valeurs élevées
 d'un dictionnaire de normes
 aux valeurs les plus basses
 d'un autre dictionnaire de
 normes;

49 / 51 ... pour indiquer qu'on souhaite
 opposer la catégorie 49 à la
 catégorie 51 du même diction-
 naire ou de deux dictionnaires
 différents;

1-9 21-29 / 10-20 30-40... pour indiquer qu'on souhaite opposer
 les catégories 1 à 9 et 21 à 29 aux catégories
 10 à 20 et 30 à 40 du même dictionnaire de
 catégories ou de deux dictionnaires de
 catégories différents;

49 / 41-70 ... pour indiquer qu'on souhaite
 opposer la catégorie 49 d'un
 dictionnaire de catégories aux
 valeurs les plus élevées d'un
 dictionnaire de normes.

Ce qui figure à gauche de la barre oblique (/) concerne le premier ensemble sémantique (SDICTS1=) (catégories ou ensembles de catégories ou bornes d'un dictionnaire de normes). Ce qui figure à droite de la barre oblique concerne le second ensemble sémantique (SDICTS2=).

C. Exemples

1. Exemple de fichier d'exécution

Contenu du fichier "cdwjuxt1.bat":

```
call protan.bat cdwjuxt rifrimem.lis rifrimem.dwj
```

2. Exemples de fichiers de paramètres et options

Contenu du fichier de paramètres et options "rifrimem.dwj":

```
context=1, punch=1, stat=1,  
swords='d:\protan\tests\laval\rifrrws2.wds',  
sdicts1='d:\protan\data.ind\ucl\uclch.dts',  
subdict1='uclem',  
sdict2='d:\protan\data\data.ind\dimabdch.dts',  
stemp='d:\protan\tests\laval\*.tmp',  
spunch='d:\protan\tests\laval\rifrimem.pcj';  
41-70 / 41-70
```

On demande à voir les phrases du fichier WORDS "rifrrws2.wds" où on trouve à la fois des occurrences de mots fort imagés (avec des poids d'imagerie entre 41 et 70, c'est-à-dire, la moitié supérieure de l'échelle d'intensité d'imagerie) et fort chargés en émotions (ayant des poids d'émotionnalité entre 41 et 70).

Le dictionnaire d'imagerie est le DIMAB; ce dictionnaire ne comporte pas de sous-dictionnaire; il n'est donc pas nécessaire de préciser le nom (SUBDICT2=) du sous-dictionnaire.

Le nom du dictionnaire d'émotionnalité est le dictionnaire UCL qui, lui, compte trois sous-dictionnaires différents, EV, AC, et EM; il faut donc préciser par l'option <SUBDICT1='uclem'>, le nom de la dimension sémantique qu'on retient dans le dictionnaire UCL.

D. Préparation des données

Le fichier WORDS doit avoir été créé (lors de l'étape CSCUT) au moyen de l'option "NARR=1" de manière à recréer l'ordre original des mots demandé par l'option "CONTEXT=1".

E. Autres informations

Le format de perforation du fichier de type PUNCHj créé par CDWJUXT est le suivant:

```
PUNCHJ  
créé par CDWJUXT  
(LRECL = 80)
```

1er enregistrement:

```
col 1 à 8 mot-clé "*PUNCHJ~"
```

9 à 16	identification du fichier de sortie
17 à 24	date de création du fichier de sortie
25 à 32	heure de ...(idem)
33 à 40	identification du fichier WORDS en entrée
41 à 48	date de création du fichier WORDS en entrée
49 à 56	heure de ...(idem)
57 à 64	identification du premier fichier DICTS en entrée (ou du premier ensemble de catégories)
65 à 72	date de création du premier fichier DICTS en entrée
73 à 80	heure de ...(idem)

2er enregistrement:

col 1 à 56	vierges
57 à 64	identification du second fichier DICTS en entrée (ou du second ensemble de catégories)
65 à 72	date de création du second fichier DICTS en entrée
73 à 80	heure de ...(idem)

3me enregistrement:

col 1 à 5	mot-clé "JUXT~"
6 à 10	mot-clé "~POPT"
11 à 15	nombre d'options de traitement fournies à CSCUT
16 à 20	mot-clé "~GCOM"
21 à 25	nombre de lignes de commentaires généraux
26 à 30	mot-clé "~OPT"
31 à 35	nombre de lignes définissant les deux ensembles de catégories (d1 et d2)
36 à 40	mot-clé "~LREQ"
41 à 45	nombre de lignes définissant la requête
46 à 50	mot-clé "~NSEG"
51 à 55	nombre de segments générés
56 à 80	vierges

4me enregistrement:

col 1 à 64	le commentaire principal du fichier SOURCE (celui de l'enregistrement *SOURCE)
65 à 80	vierges

les POPT enregistrements suivants:

col 1 à 80	une option sous la forme XXX=YYY;
------------	-----------------------------------

les GCOM enregistrements suivants:

col 1 à 80	un commentaire général
------------	------------------------

les OPT enregistrements suivants:

col 1 à 80	une option sous la forme XXX=YYY;
------------	-----------------------------------

les LREQ enregistrements suivants:

col 1 à 80	une ligne de la requête
------------	-------------------------

enregistrements suivants (NSEG enregistrements): chaque variable occupe 9 positions et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

- numéro de segment (variable SEG)
- nombre total de mots dans ce segment (variable WT)
- nombre de mots différents dans ce segment (variable WD)
- nombre de phrases où il y a juxtaposition dans ce segment (variable NJ)
- nombre de phrases où au moins une catégorie de l'ensemble d1 est présente dans ce segment (variable FD1)

- idem pour l'ensemble d2 (variable FD2)
- SQRT RATE pour ce segment (variable RATE), représenté par un nombre fractionnaire contenant trois décimales précédées du point décimal,
... et passage à l'enregistrement suivant pour la 2ième observation.

Le SQRT Rate est l'indice final qui synthétise toutes les co-occurrences recherchées selon la formule:

$$\text{SQRT } ((\text{NJ}/(\text{FD1}+\text{FD2})) \times 1.000)$$

où NJ est le nombre de phrases (par segment) avec juxtaposition,
où FD1 est le nombre de phrases où au moins une catégorie de l'ensemble sémantique d1 est présent,
et où FD2 est le nombre de phrases où au moins une catégorie de l'ensemble sémantique d2 est présent.

F. Remarques

Aucune

G. Messages propres au programme CDWJUXT

```
*CDWJ841I only one dictionary will be used
*CDWJ842S option "NARR=1" not provided to "CSCUT"
          - "WORDS" file cannot be processed
*CDWJ843S no word has been tagged by dictionaries
*CDWJ844S invalid category
*CDWJ845S invalid '-' specification
*CDWJ846S invalid '/' specification
*CDWJ847S syntax error in request definition
*CDWJ848S duplicate category
*CDWJ849S category undefined in dictionary
*CDWJ850S one dictionary - the 2 cat. series are in overlay
```

X. Le programme CDWLOOK

H. But du programme

Le but du programme CDWLOOK est de comparer les mots d'un texte aux mots d'un dictionnaire de catégories ou de normes et de fournir une estimation chiffrée, sous forme de différents scores, du degré de présence de la ou des catégories du dictionnaire dans le texte. Accessoirement, ces scores peuvent être perforés, par segment ou par catégories, en sorte de pouvoir être soumis, hors système, comme fichiers de données à diverses analyses statistiques.

I. Exécution

1. Appel du programme

```
... CDWLOOK fn1 fn2 fn3
```

a. où fn1 est le filename d'un fichier dont le filetype est CDWLOOK. Ce fichier "fn1 CDWLOOK" donnera son nom "fn1" aux fichiers de filetype "PUNCHM" ou "SASx" (soit "SASCAT", "SASCATOB", "SASNOR", "SASCATSQ", "SASNORSQ", "SASCATNA" ou "SASNORNA") (les résultats perforés) si de tels fichiers sont demandés dans les options contenues dans le fichier "fn1 CDWLOOK".

b. où fn2 est le filename du fichier de filetype "DICTS", c'est-à-dire le nom du dictionnaire utilisé pour l'analyse du corpus.

c. où fn3 est le filename du fichier de filetype "WORDS" créé par CSCUT, CRWSTRIP, CWEDIT, etc. Ce fichier "fn3 WORDS" contient le corpus faisant l'objet de l'analyse.

2. Entrées/sorties: Le programme CDWLOOK ne crée pas de sortie sinon, sur demande, un fichier de type "PUNCHM" ou "SASx" (soit "SASCAT", "SASCATOB", "SASNOR", "SASCATSQ", "SASNORSQ", "SASCATNA" ou "SASNORNA") qui contient les résultats perforés. En entrée, le programme demande un fichier de paramètres (fn1), un dictionnaire (de filetype "DICTS"), et un corpus (de filetype "WORDS").

3. Paramètres et options:

STAT = Par défaut ("STAT = 0"), il n'y a pas impression des statistiques de tri qui accompagnent les fichiers de type "WORDS". "STAT = 1" déclenche l'impression de ces statistiques, c'est-à-dire, par segment et pour le total, le nombre de mots différents et le nombre total de mots.

WORDT = On peut demander que soient imprimés les mots des catégories pour lesquelles le programme a trouvé une correspondance avec un mot du texte en cours d'analyse. Par défaut ("WORDT = 0"), cette impression ne se fait pas. Mais si "WORDT = n" ou si "WORDT = 255", il y aura impression des n ou de toutes les catégories, pour n et 255

respectivement, pour lesquelles une correspondance a été trouvée. Et si on a choisi "WORDT = n", c'est-à-dire si WORDT est plus grand que zéro et plus petit que 255, il faudra en plus énumérer, en fin du fichier "fn1 CDWLOOK", les numéros d'ordre de ces n catégories, séparés par un ou plusieurs blancs (~).

De gauche à droite sur la page de liste imprimée, on trouve les rubriques suivantes: / SEGMENT / TEXT WORD / DICT WORD / CATEGORY / FREQ / SQRT RATE FREQ /.

Ainsi, on va observer par exemple, que dans le segment no. 3 (SEGMENT) composé de 161 mots, le mot "allumer" dans le texte (TEXT WORD) a été repris dans les comptages parce qu'il y a, dans le dictionnaire d'imagerie régressive DIRE, un mot "allum." (DICT WORD) --qui se trouve être un radical dans ce cas-ci-- assigné à la catégorie no. 48 d'imagerie icarienne (CATEGORY). Ce TEXT WORD apparaît dans ce SEGMENT avec une fréquence de 1. Ceci donne lieu au calcul du taux (SQRT RATE FREQ) de fréquence de ce mot, taux qui est la racine carrée du rapport (multiplié par 1000) de la fréquence du mot au nombre total de mots dans ce segment, soit:

$$\text{SQRT } [(1/161)*1000] = 2.492$$

puisqu'il y a 161 mots dans ce segment.

CATLOW =

CATUP = Ces deux options ne fonctionnent (et n'ont de sens) qu'avec les dictionnaires de normes, et ont un effet sur les autres options PRINTT, CONT, PRINTNA, et PUNCH. Si CATLOW est inférieur ou égal à la catégorie la plus basse du dictionnaire, ou si CATUP est supérieur ou égal à la catégorie la plus haute du dictionnaire (de norme), CATLOW et CATUP prennent respectivement ces valeurs-là. Ceci constitue le défaut.

Mais si CATLOW est supérieur à la catégorie la plus basse du dictionnaire, et/ou si CATUP est inférieur à la catégorie la plus élevée (du dictionnaire de normes toujours), alors, le calcul du taux de présence de la catégorie dans le texte commencera à partir de cette borne inférieure "relevée" et/ou se limitera à cette borne supérieure "abaissée".

Par exemple, le dictionnaire de normes d'imagerie DIMAB couvre les catégories 10 à 70; par défaut, CATLOW=10 et CATUP=70. Si on souhaite limiter les effets du dictionnaire aux normes d'imagerie élevée, soit les catégories 50 à 70, on définira CATLOW= 50 et CATUP=70. Mais si on souhaite différencier l'imagerie faible (10 à 30), l'imagerie moyenne (31 à 50), et l'imagerie élevée (51 à 70), on effec-

tuera trois exécutions différentes du programme, sur le même fichier "WORDS" et avec le même dictionnaire DIMAB, chaque fois avec des valeurs d'option différentes, soit "CATLOW=10,CATUP=30", puis, lors de l'exécution suivante "CATLOW=31, CATUP=50", et enfin "CATLOW=51, CATUP=70".

SCORE = Cette option ne fonctionne (et n'a de sens) qu'avec les dictionnaires de normes, en affectant les résultats des options PRINTT, CONT, et PUNCH.

Par défaut, SCORE = 'AVERTAG', qui est un score moyen. Au numérateur de ce score moyen figure la somme des produits de la valeur de chaque catégorie par le nombre total de mots trouvés dans le segment en correspondance avec chaque catégorie; et on trouve au dénominateur la somme des fréquences des mots trouvés en correspondance avec chaque catégorie dans le segment considéré.

SCORE = 'AVERTOT' est aussi un score moyen. La somme figurant au numérateur est celle calculée pour SCORE = 'AVERTAG', mais ici, cette somme est divisée par le nombre total de mots dans le segment considéré.

CATSEL = Pour les dictionnaires de catégories et seulement pour ceux-là, l'option CATSEL permet de sélectionner, à condition de les énumérer en fin de fichier de paramètres après les numéros des catégories à énumérer en rapport avec WORDT, les catégories sur lesquelles les options PRINTT, CONT, PRINTNA, et PUNCH (voir plus loin) devront agir. En d'autres mots, CATSEL va sélectionner les catégories pour lesquelles des statistiques descriptives (élémentaires) doivent être imprimées et/ou perforées. Par défaut, "CATSEL = 0" et aucune catégorie n'est sélectionnée; dès ce moment, les options PRINTT, CONT, PRINTNA, et PUNCH sont ignorées. Si "CATSEL = n", on demande que n catégories soient analysées et/ou perforées, et ces catégories doivent être énumérées en fin de fichier. Si "CATSEL = 255", on demande que toutes les catégories soient analysées et/ou perforées, sans qu'il soit nécessaire dans ce cas-ci de les énumérer. Dès le moment où "CATSEL = n" ou "CATSEL = 255", l'une des options PRINTT, CONT, PRINTNA, ou PUNCH au moins doit être activée.

PRINTT = "PRINTT = 1" permet l'impression (inhibée par le défaut "PRINTT = 0"), par segment, des taux de fréquence et de densité de chaque catégorie. Dans le cas d'un dictionnaire de normes, il y a en outre l'impression des taux moyens de fréquence et de densité pour le segment considéré. De gauche à droite de la page, on trouve les rubriques /CATEGORY / FREQ / SQRT RATE FREQ / DENS / SQRT RATE DENS/.

Par exemple, dans le segment no. 2 du fichier "INTRUSE WORDS" traité selon le dictionnaire (catégoriel) d'imagerie régressive DIRE, la catégorie no. 48 (CATEGORY) est évoquée 25 fois (FREQ). Comme il y a 356 mots dans ce segment, le taux de fréquence (SQRT RATE FREQ) de cette catégorie est égal à:

$$\text{SQRT} [(25/356)*1000] = 8.38$$

Par ailleurs, la valeur de densité (DENS) de cette catégorie n'est que de 16. Ceci nous indique que la fréquence de 25 constatée plus haut est due à 16 mots différents. Et le taux de densité (SQRT RATE DENS) de cette catégorie est alors égal à:

$$\text{SQRT} [(16/356)*1000] = 6.704$$

La valeur de densité est toujours, et nécessairement, inférieure, ou au plus égale, à celle de fréquence. Mais surtout, la valeur de densité peut être un indice plus sensible de la présence d'une catégorie que celle de fréquence. Pour prendre un exemple extrême, on pourrait imaginer que le SQRT RATE FREQ de 8.380 soit dû à un seul mot répété 25 fois dans le segment, tout comme il pourrait être dû à 25 mots différents présents chacun une seule fois dans le segment. Mais ces deux situations ne sont pas comparables.

Dans le cas d'un dictionnaire de normes, on trouve, en outre, des taux moyens de fréquence et de densité (AVERAGE SCORE FREQ et AVERAGE SCORE DENS). La formule de calcul de ces taux est bien sûr déterminée par le choix des valeurs (AVERTAG ou AVERTOT) de l'option SCORE.

Pour le segment no. 2 du même texte "INTRUSE" traité selon le dictionnaire de normes d'imagerie DIMAB, si SCORE = 'AVERTAG', les valeurs de taux moyens sont de:

$$\begin{aligned} \text{AVERAGE SCORE FREQ} &= 46.846 \\ \text{AVERAGE SCORE DENS} &= 44.677 \end{aligned}$$

et si SCORE = 'AVERTOT', ces valeurs sont de:

$$\begin{aligned} \text{AVERAGE SCORE FREQ} &= 5.132 \\ \text{AVERAGE SCORE DENS} &= 3.890 \end{aligned}$$

CONT = "CONT = 1" permet l'impression, par catégorie, de quelques statistiques descriptives sur les valeurs SQRT RATE FREQ et SQRT RATE DENS (pour les dictionnaires de catégories) et sur les valeurs AVERAGE SCORE FREQ et AVERAGE SCORE DENS (pour les dictionnaires de normes). Le défaut "CONT = 0" inhibe cette impression.

De gauche à droite sur la page, on trouve les rubriques / SEG / TEXT WORDS TOTAL / TEXT WORDS DIFF. / FREQ / SQRT RATE FREQ / DENS / SQRT RATE DENS / pour les dictionnaires de catégories. Pour les dictionnaires de normes, ces rubriques deviennent / SEG / TEXT WORDS TOTAL / TEXT WORDS DIFF. / WORDS TAGGED TOTAL / WORDS TAGGED DIFF. / AVER. SCORE FREQ / AVER. SCORE DENS /.

Par exemple, pour le texte "INTRUSE" traité selon le dictionnaire DIRE (catégoriel), les options "CATSEL = 1, CONT = 1" donnent les résultats suivants --la catégorie sélectionnée est celle d'imagerie icarienne, no. 48, couvrant les sous-catégories 24 à 29, soit "Monter", "Haut", "Descendre", "Profondeur", "Feu", et "Eau"--:

**		"INTRUSE "	**	
SCORES FOR CATEGORY BASED DICTIONARY:				
CATEGORY 48 IMAGERIE ICARIENNE :....24-29				
SEG SQRT RATE FREQ...			SQRT RATE DENS	
1	1.486	...	1.486	
2	1.676	...	1.676	
3	2.195	...	2.195	
4	2.528	...	2.528	
5	3.525	...	3.525	
Mean	2.282		2.282	
sd	0.723		0.723	
med	2.195		2.195	

On reconnaît dans l'encadré ci-dessus les 5 segments (SEG) du texte, les valeurs SQRT RATE FREQ et SQRT RATE DENS pour chacun des 5 segments, ainsi que les moyennes, sigmas, et médianes.

WORDNT = "WORDNT = n" donne la liste des mots non repris par le dictionnaire, mais dont les n premiers caractères sont identiques à ceux d'un des mots existant dans le dictionnaire **et** dans le texte. Pour chacun des mots non repris, on trouve une indication du segment où il se situe et sa fréquence. "WORDNT = 0" inhibe l'impression de cette liste (défaut).

Cette option permet de repérer des mots susceptibles d'être intégrés utilement dans un dictionnaire déjà constitué. Ainsi, dans le texte "INTRUSE", il est signalé le mot "actif": Ce mot ressemble en effet par ses trois premiers caractères ("WORDNT = 3") à plusieurs mots du dictionnaire DIMAB, tels que "acte, actes, action, activité, actrice", et pourrait être inclus dans une nouvelle version du dictionnaire si elle devait être entreprise.

PRINTNA= Il est impossible de reconstituer un texte dans son ordre séquentiel original à partir d'un fichier de type WORDS. Un fichier

de ce type contient en effet essentiellement, en ordre alphabétique, les mots différents du texte avec leurs indications de segment et de fréquence. Cependant, si l'option "NARR = 1" a été incluse dans les options du programme CSCUT¹², il est possible quand même, lorsqu'on est dans le programme CDWLOOK, de rétablir l'ordre séquentiel initial du texte.

Cet ordre est rétabli, et le texte est imprimé, en activant l'option "PRINTNA". Le défaut de cette option est "PRINTNA = 0". "PRINTNA = 1" numérote tous les items du corpus, tandis que "PRINTNA = 2" ne numérote que les mots (c'est-à-dire pas les ponctuations, marqueurs, etc.) en correspondance avec PUNCH = 3 et PUNCH = 4 et en concordance avec PRINT = 2 de CSCUT.

L'option PRINTNA est complétée par sept autres options qui permettent de préciser la forme et la quantité d'informations qu'on souhaite imprimer. Ce sont les options LL, TAGLOW/UP, SEGLOW/UP, et SEQLOW/UP (voir plus loin la section "Remarques" pour des commentaires sur l'utilité de PRINTNA).

A noter que les comparaisons entre dictionnaire et corpus se font sur la génération la plus récente du fichier WORDS (strippé). Par contre, l'impression via PRINTNA se fait en utilisant le niveau 0 du fichier WORDS (c'est-à-dire la sortie non strippée de CSCUT).

LL = Puisque PRINTNA imprime un texte, il est utile de pouvoir gérer le format de cette impression. L'option LL gère la longueur de ligne du texte, c'est-à-dire le nombre de caractères par ligne imprimée. Par défaut, LL est égal à 80; on peut le faire varier de 45 à 132.

TAGLOW =

TAGUP=

Ces deux options sont les bornes inférieure et supérieure des numéros de séquence des mots retenus par le dictionnaire utilisé. Un texte peut être long, et, si on ne met pas de barrière, tout le texte sera imprimé par l'option "PRINTNA = 1". Le défaut de TAGLOW est 0, c'est-à-dire le début du texte, tandis que le défaut de TAGUP est le plus grand nombre de 8 chiffres, soit 99.999.999 mots assignés!

Il est vraisemblable que, la plupart du temps, seules des portions de texte seront imprimées, celles où l'on veut voir en détail comment se fait l'"étiquetage" par le dictionnaire.

SEGLOW =

SEGUP =

Ces deux options sont, elles, les bornes inférieure et supérieure des numéros de

¹² "NARR = 1" ajoute aux indications contenues dans le fichier WORDS, des attributs de séquence pour chaque occurrence de mot.

segment. Plutôt que d'imprimer une portion de texte en fonction des numéros de séquence des mots retenus par le dictionnaire, on peut préférer imprimer une portion de texte correspondant à un ou quelques segments successifs. Le défaut de SEGLOW est 1, le premier segment; le défaut de SEGUP est le plus grand nombre de 5 chiffres, soit 99.999, le nombre maximum de segments que le système peut accepter.

SEQLOW =

SEQUP = Enfin, ces deux options sont les bornes inférieure et supérieure des numéros de séquence des mots du texte. Encore une fois, plutôt que d'imprimer tout le texte, on peut choisir de n'imprimer que la portion de texte correspondant aux mots un tel à un tel. Le défaut de SEQLOW est 1, le premier mot du texte; le défaut de SEQUP est le plus grand nombre de 8 chiffres, 99.999.999, le nombre maximum de mots que le système puisse traiter.

SUBDICT=

Nécessaire pour identifier un sous-dictionnaire d'un dictionnaire de normes au cas où il y a plusieurs sous-dictionnaires.

SWORDS=

Cette option est nécessaire pour l'utilisation de PROTAN en PC, pour fournir au programme le nom et le chemin du fichier WORDS. Par exemple, SWORDS = 'd:....CAMUSrws.wds'.

SDICTS =

Cette option est nécessaire pour l'utilisation de PROTAN en PC, pour fournir au programme le nom (et le chemin éventuellement) du fichier DICTS. Par exemple, SDICTS = 'd:....CAMUSdch.dts'.

SPUNCH=

Option nécessaire pour l'utilisation de PROTAN en PC, pour fournir au programme le nom et le chemin du fichier de type PUNCHM. Par exemple, SPUNCH = 'd:....CAMUSdwl.pcm'.

PUNCH =

"PUNCH = 0", le défaut, ne donne lieu à aucune perforation de résultats. "PUNCH = 1" ou "PUNCH = 2" perforent les résultats imprimés par l'option CONT (à l'exception des moyennes, sigmas, médianes, et titres des rubriques) sous la forme d'un fichier de type "PUNCHM". Ce dernier fichier de type "PUNCHM" peut lui-même être converti, par l'option SAS de la procédure d'appel, en un fichier perforé de type SASx (soit SASCAT, SASCATOB, ou SASNOR). Si "PUNCH = 1", ces résultats sont perforés par segment, et si "PUNCH = 2", ils sont perforés par catégories.

Prenons l'exemple du texte "JUAREZ" traité selon le dictionnaire DIRE. Si les valeurs des options sont "CATSEL = 1, PUNCH = 1;" pour la catégorie sélectionnée no. 49 (processus primaires), le fichier de type PUNCHM produit se présente sous la forme:

```
*PUNCHM JUAREZ 22/06/8814:54:10JUAREZ 22/06/8814:53:52
                                DIRE 23/09/8714:33:48
CAT  POPT    6 GCOM    4 OPT    1 NSEG  11 NCAT    1

CTRL=                211;
SEGT='W';
SENT='.];?:';
BRKM=                1;
BRKB='U  ';
NARR=                1;
PROC BY CRWSTRIP "JUAREZ  " (22/06/88 14:53:52)
                  "FRSTRP01" (14/03/88 10:27:40)
PROC BY CSCUT    "JUAREZ  " (22/06/88 14:53:32)
  " HISTOIRE DE ROSENDO JUAREZ " TIRE DE J.L. BORGES ,
    IN "LE RAPPORT DE BRODIE", PP. 39-48.
CATSEL=          1;
49
  1      211      132                10  6.884          9  6.531
  2      211      131                12  7.541          10  6.884
  3      211      118                21  9.976          19  9.489
  4      211      135                20  9.736          19  9.489
  5      211      134                16  8.708          12  7.541
  6      211      120                 9  6.531           8  6.157
  7      211      122                 9  6.531           8  6.157
  8      211       99                 3  3.771           3  3.771
  9      211      118                 6  5.333           6  5.333
 10      211      123                10  6.884          10  6.884
 11      211      120                20  9.736          16  8.708
```

Ce fichier de type "PUNCHM" avec option "PUNCH= 1" se lit donc de haut en bas, les 7 colonnes correspondant aux intitulés / SEG / TEXT WORDS TOTAL / TEXT WORDS DIFF. / FREQ / SQRT RATE FREQ / DENS / SQRT RATE DENS / qu'on trouve en tête des listes imprimées (par l'option "CONT = 1"). Les catégories (variables) y varient plus vite que les segments (observations). Si "PUNCH = 1" est remplacé par "PUNCH = 2", les autres options restant inchangées, le fichier de type "PUNCHM" produit devient:

```

*PUNCHM JUAREZ 22/06/8814:54:21JUAREZ 22/06/8814:53:52
                                DIRE 23/09/8714:33:48
CATOB POPT 6 GCOM 4 OPT 1 NSEG 11 NCAT 1

CTRL= 211;
SEGT='W';
SENT='.];?:';
BRKM= 1;
BRKB='U ';
NARR= 1;
PROC BY CRWSTRIP "JUAREZ " (22/06/88 14:53:52)
                "FRSTRP01" (14/03/88 10:27:40)
PROC BY CSCUT "JUAREZ " (22/06/88 14:53:32)
" HISTOIRE DE ROSENDO JUAREZ " TIRE DE J.L. BORGES ,
    IN "LE RAPPORT DE BRODIE", PP. 39-48.
CATSEL= 1;
49
49      211      132      10 6.884      9 6.531
        211      131      12 7.541     10 6.884
        211      118      21 9.976     19 9.489
        211      135      20 9.736     19 9.489
        211      134      16 8.708     12 7.541
        211      120      9 6.531      8 6.157
        211      122      9 6.531      8 6.157
        211      99       3 3.771      3 3.771
        211      118      6 5.333      6 5.333
        211      123     10 6.884     10 6.884
        211      120     20 9.736     16 8.708

```

Il s'agit donc toujours du même fichier de type "PUNCHM", mais transposé, les variables y devenant des observations, et les observations y devenant des variables. (Comme il n'y a qu'une seule catégorie considérée, les deux exemples précédents ne diffèrent presque pas).

Les deux autres PUNCH, "PUNCH = 3" et "PUNCH = 4", perforent les résultats dans l'ordre séquentiel initial du texte. L'intérêt de ce type d'opérations est de permettre d'entrer le produit de ce PUNCH directement dans des analyses de type séquentiel (Bakeman & Gottman, 1986; Sarbin, 1986) ou de séries temporelles (Fuller, 1976; Hogenraad & Bestgen, 1989).

"PUNCH = 3" produit un fichier de résultats perforés sous la forme:

```

*PUNCHM JUAREZ 22/06/8814:54:32JUAREZ 22/06/8814:53:52
                                DIRE 23/09/8714:33:48
CATSQ POPT      6 GCOM      4 OPT      1 NSEG      11 NCAT      1 NTAG      137

CTRL=          211;
SEGT='W';
SENT='.];?:';
BRKM=          1;
BRKB='U  ';
NARR=          1;
PROC BY CRWSTRIP "JUAREZ  " (22/06/88 14:53:52)
                  "FRSTRP01" (14/03/88 10:27:40)
PROC BY CSCUT    "JUAREZ  " (22/06/88 14:53:32)
" HISTOIRE DE ROSENDO JUAREZ " TIRE DE J.L. BORGES ,
    IN "LE RAPPORT DE BRODIE", PP. 39-48.
CATSEL=          1;
  49
    1      11 49      1      23 49      1      29 49      1      31 49
...
    .....
    .....
11      2321 49      12      2322 0

```

On observe que les mots retenus par le dictionnaire sont présents, et en séquence: Dans le premier segment, et pour la seule catégorie no. 49, les mots en position 11, 23, 29, 31, etc... ont été retenus (c'est-à-dire ont une valeur dans la catégorie 49 du dictionnaire), et ainsi de suite, jusqu'au onzième segment qui se termine par le mot en position 2.321. (Le mot en position 2.322, segment 12, est une valeur fantôme destinée à connaître le nombre total de segments et de mots dans le texte.)

Enfin, "PUNCH = 4" produit un fichier de résultats perforés sous la forme:

```

*PUNCHM JUAREZ 22/06/8814:54:48JUAREZ 22/06/8814:53:52
                                DIRE 23/09/8714:33:48
CATNA POPT      6 GCOM      4 OPT      1 NSEG      11 NCAT      1 NTAG 2321

CTRL=          211;
SEGT='W';
SENT='.];?:';
BRKM=          1;
BRKB='U  ';
NARR=          1;
PROC BY CRWSTRIP "JUAREZ  " (22/06/88 14:53:52)
                  "FRSTRP01" (14/03/88 10:27:40)
PROC BY CSCUT    "JUAREZ  " (22/06/88 14:53:32)
  " HISTOIRE DE ROSENDO JUAREZ " TIRE DE J.L. BORGES ,
    IN "LE RAPPORT DE BRODIE", PP. 39-48.
CATSEL=          1;
  49
IL              1          1 0      DEVAIT          1          2 0
E3TRE           1          3 0      ONZE            1          4 0
HEURES          1          5 0      DU              1          6 0
SOIR            1          7 0      J               1          8 0
E1TAIS          1          9 0      ENTRE1          1         10 0
DANS            1         11 49      LE              1         12 0
BISTROT         1         13 0      QUI             1         14 0
..              ..          ... .      ....          ..          ... .
..              ..          ... .      ....          ..          ... .
JE              11         2309 0      ME            11         2310 0
SUIS            11         2311 0      INSTALLE1     11         2312 0
ICI             11         2313 49      SAN          11         2314 0
TELMO          11         2315 0      A             11         2316 0
TOUJOURS       11         2317 49      E1TE1        11         2318 0
UN             11         2319 0      QUARTIER       11         2320 0
TRANQUILLE     11         2321 49

```

Ici, tout le texte se trouve en séquence --le fichier se lit 'partie gauche, partie droite, partie gauche suivante, partie droite suivante, ... pour former la phrase "IL DEVAIT E3TRE ONZE HEURES DU SOIR ..."). On voit par exemple que le mot en position 11 est assigné à la catégorie no. 49 du dictionnaire, ce qu'on savait par le tableau précédent, le contexte en moins.

En résumé, sont communs aux dictionnaires de normes et aux dictionnaires de catégories, les six options STAT, WORDT, PRINTT, WORDNT, CONT, PRINTNA (et ses options satellites), et PUNCH. Sont spécifiquement conçues et formulées pour les dictionnaires de normes, les quatre options CATLOW, CATUP, SCORE, et SUBDICT. Enfin, l'option CATSEL est conçue et formulée pour le dictionnaire de catégories seulement.

Rappel: Les numéros des catégories pour WORDT (d'abord) et CATSEL (ensuite) se placent après la liste des options. Ces numéros sont séparés par un ou plusieurs blancs (~).

J. Exemples

1. Exemple de procédure d'appel

a. " ... CDWLOOK INTRUSE1 DIMAB INTRUSE"

Un fichier "INTRUSE1 CDWLOOK" avec les paramètres est appelé en premier lieu, suivi du dictionnaire servant à analyser le corpus,

ici, le dictionnaire "DIMAB DICTS", le corpus étant contenu dans le troisième fichier appelé "INTRUSE WORDS".

2. Exemples de fichiers de paramètres et options

a. Contenu du fichier "INTRUSE1 CDWLOOK" (entre < et > non compris):

```
<
STAT = 1, WORDT= 255, CATLOW= 10, CATUP= 70, SCORE = 'AVERTAG',
PRINTT = 1, CONT = 1, WORDNT = 6, PUNCH = 1;
>
```

Certaines de ces options pourraient être supprimées puisqu'on demande le défaut de CATLOW, CATUP, et SCORE. On aurait pu écrire:

```
<
STAT = 1, WORDT = 255, PRINTT = 1, CONT = 1, WORDNT = 6, PUNCH =
1;
>
```

On demande donc

- (1) que les statistiques de tri soient imprimées,
- (2) que soient imprimées toutes les catégories pour lesquelles une correspondance a été trouvée avec un mot du texte,
- (3) que soient imprimés les taux des fréquences et densités par segment, d'ailleurs pour toute l'étendue du dictionnaire, soit des catégories 10 à 70,
- (4) que soient fournies les statistiques descriptives sur les valeurs AVERAGE SCORE FREQ et AVERAGE SCORE DENS, c'est-à-dire moyennes, sigmas, et médianes,
- (5) que soit imprimée la liste des mots qui n'ont pas été assignés à une catégorie mais qui pourraient l'être éventuellement en raison de leurs 6 premiers caractères --puisque WORDNT = 6--, et enfin
- (6) que les résultats soient perforées par segment.

b. Contenu du fichier "INTRUSE2 CDWLOOK" (entre < et > non compris):

```
<
STAT = 1, CATSEL = 1, CONT = 1, PUNCH = 1;
48
>
```

Outre les statistiques de tri, on demande que l'analyse (CONT) et la perforation (PUNCH) par segment ne portent que sur une seule catégorie (CATSEL = 1) désignée en fin de fichier, la catégorie 48. Les autres options sont prises par défaut.

K. Préparation des données

Aucune.

L. Autres informations

Le format de perforation du fichier de type "PUNCHM" que crée CDWLOOK est caractérisé par un LRECL de 80. Ce format diffère selon que l'on est en présence d'un dictionnaire de catégories ou de normes et que PUNCH a les valeurs 1, 2, 3 ou 4. L'encadré qui suit contient la description de ce fichier:

PUNCHM
créé par CDWLOOK
(LRECL = 80)

1er enregistrement:

col 1 à 8	mot-clé "*PUNCHM~"
9 à 16	identification du fichier de sortie
17 à 24	date de création du fichier de sortie
25 à 32	heure de ...(idem)
33 à 40	identification du fichier WORDS en entrée
41 à 48	date de création du fichier WORDS en entrée
49 à 56	heure de ...(idem)
57 à 64	identification du fichier DICTS en entrée
65 à 72	date de création du fichier DICTS en entrée
73 à 80	heure de ...(idem)

2me enregistrement:

col 1 à 5	mot-clé dépendant du type de dictionnaire et de la valeur attribuée à l'option PUNCH:
	"CAT~~" : dict. de catégories et PUNCH = 1
	"CATOB" : dict. de catégories et PUNCH = 2
	"CATSQ" : dict. de catégories et PUNCH = 3
	"CATNA" : dict. de catégories et PUNCH = 4
	"NOR~~" : dict. de normes et PUNCH = 1 ou 2
	"NORSQ" : dict. de normes et PUNCH = 3
	"NORNA" : dict. de normes et PUNCH = 4
6 à 10	mot-clé "~POPT"
11 à 15	nombre d'options de traitement fournies à CSCUT
16 à 20	mot-clé "~GCOM"
21 à 25	nombre de lignes de commentaires généraux
26 à 30	mot-clé "~~OPT"
31 à 35	nombre d'options de traitement fournies au programme ayant généré ce fichier
36 à 40	mot-clé "~NSEG"
41 à 45	nombre de segments générés
46 à 50	mot-clé "~NCAT"
51 à 55	nombre de catégories sélectionnées
56 à 60	mot-clé "~NTAG"; uniquement pour PUNCH = 3 (CATSQ ou NORSQ) et PUNCH = 4 (CATNA ou NORNA)
61 à 70	nombre de mots retenus (pour toutes les catégories sélectionnées) + 1, pour PUNCH = 3 (CATSQ ou NORSQ);
	nombre de mots retenus (pour toutes les catégories sélectionnées) plus le nombre de mots non retenus, pour PUNCH = 4 (CATNA ou NORNA);
56 à 70	vierges pour PUNCH = 1 ou 2 (CAT, CATOB ou NOR)
71 à 80	vierges

3me enregistrement:

col 1 à 64	le commentaire principal du fichier SOURCE (celui de l'enregistrement *SOURCE)
65 à 80	vierges

les POPT enregistrements suivants:

```

col 1 à 80    une option sous la forme XXX=YYY;

les GCOM enregistrements suivants:
col 1 à 80    un commentaire général

les OPT enregistrements suivants:
col 1 à 80    une option sous la forme XXX=YYY;

enregistrements suivants (au nombre de (NCAT+19)/20):
col 1 à 80    par groupe de 4 colonnes, les NCAT catégories
               sélectionnées

```

A. Pour un dictionnaire de catégories et PUNCH = 1 ("CAT~~")

enregistrements suivants (NSEG enregistrements): chaque variable occupe 8 positions et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

- numéro du segment en cours (variable SEG)
- nombre total de mots dans le segment en cours (variable WT)
- nombre de mots différents dans le segment en cours (variable WD)

...l'ensemble suivant se répète autant de fois qu'il y a de catégories sélectionnées):

- fréquence de la première catégorie pour le segment en cours (variable F1)
- SQRT RATE FREQ de la première catégorie pour le segment en cours (variable RF1)
- densité de la première catégorie pour le segment en cours (variable D1)
- SQRT RATE DENS de la première catégorie pour le segment en cours (variable RD1).

Si NCAT > 1, on trouve en plus les variables F2, RF2, D2, RD2,...

Et passage à l'enregistrement suivant pour la 2ième segment, etc...

Note: La fréquence et la densité sont représentées par un nombre entier et les SQRT RATE, par un nombre fractionnaire contenant trois décimales précédées du point décimal.

B. Pour un dictionnaire de catégories et PUNCH = 2 ("CATOB")

enregistrements suivants (NCAT enregistrements): chaque variable occupe 8 positions et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

- numéro de la catégorie en cours (variable CAT)

... l'ensemble suivant se répète autant de fois qu'il y a de segments...

- nombre total de mots dans le premier segment (variable WT1)

- nombre de mots différents dans le premier segment (variable WD1)
- fréquence de la catégorie en cours pour le premier segment (variable F1)
- SQRT RATE FREQ de la catégorie en cours pour le premier segment (variable RF1)
- densité de la catégorie en cours pour le premier segment (variable D1)
- SQRT RATE DENS de la catégorie en cours pour le premier segment (variable RD1).

Si NSEG > 1, on trouve les valeurs des variables WT2, WD2, F2, RF2, D2, RD2, ...

...et passage à l'enregistrement suivant pour la 2ième catégorie.

Note: La fréquence et la densité sont représentées par un nombre entier et les SQRT RATE, par un nombre fractionnaire contenant trois décimales précédées du point décimal.

C. Pour un dictionnaire de catégories et PUNCH = 3 ("CATSQ")

enregistrements suivants (NTAG enregistrements): ceux-ci sont rangées en ordre croissant des numéros de séquence des mots et des numéros de catégorie pour chaque numéro de séquence. Chaque variable occupe 8 positions et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

- numéro du segment (variable SEG)
- numéro de séquence du mot retenu (variable SE-
QUENCE)
- numéro de la catégorie (variable CAT)

...et passage à l'enregistrement suivant pour la 2ième observation, etc.

Note: La dernière observation, fictive, est caractérisée par un numéro de segment égal à NSEG+1, un numéro de séquence égal au nombre total de mots +1 et un numéro de catégorie nul.

D. Pour un dictionnaire de catégories et PUNCH = 4 ("CATNA")

enregistrements suivants (NTAG enregistrements): ils sont rangés en ordre croissant des numéros de séquence des mots et des numéros de catégorie pour chaque numéro de séquence. Chaque variable occupe 8 positions (sauf le mot -WORD- qui en occupe 20) et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

- un mot (variable WORD)
- numéro du segment (variable SEG)
- numéro de séquence du mot retenu (variable SE-
QUENCE)
- numéro de la catégorie (variable CAT)

... et passage à l'enregistrement suivant pour le 2ième segment, etc.

E. Pour un dictionnaire de normes et PUNCH = 1 ou 2 ("NOR~~")

enregistrements suivants (NSEG enregistrements): chaque variable occupe 11 positions et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

- numéro de segment (variable SEG)
- nombre total de mots dans ce segment (variable WT)
- nombre de mots différents dans ce segment (variable WD)
- nombre total de mots retenus dans ce segment (variable TAT)
- nombre de mots retenus différents dans ce segment (variable TAD)
- sommes des produits "norme * fréquence" pour ce segment (variable NORF)
- AVERAGE SCORE FREQ pour ce segment (variable AF)
- sommes des produits "norme * densité" pour ce segment (variable NORD)
- AVERAGE SCORE DENS pour ce segment (variable AD)

...et passage à l'enregistrement suivant pour le 2ième segment, etc...

Note: Les AVERAGE SCORE sont représentés par un nombre fractionnaire contenant trois décimales précédées du point décimal.

F. Pour un dictionnaire de normes et PUNCH = 3 ("NORSQ")

Cf. le point C. relatif à un dictionnaire de catégories.

G. Pour un dictionnaire de normes et PUNCH = 4 ("NORNA")

Cf. le point D. relatif à un dictionnaire de catégories.

Ce fichier de type "PUNCHM" ne doit être demandé que dans le cas où les données qu'il contient sont susceptibles de faire l'objet d'analyses statistiques. Ce fichier reprend en fait les informations données par l'option "CONT = 1".

Cependant, si l'option SAS a été renseignée dans la procédure d'appel du programme CDWLOOK, par exemple:

```
< ... CDWLOOK ... (SAS)>,
```

ce fichier de type "PUNCHM" est automatiquement converti en un fichier SAS. Ce fichier SAS est caractérisé par des types différents selon la nature du dictionnaire utilisé (normes ou catégories) et selon la valeur de l'option PUNCH. On aura donc un fichier SAS:

```

de type SASCAT: pour un dictionnaire de catégories avec PUNCH=1
de type SASNOR: pour un dictionnaire de normes avec PUNCH=1
de type SASCATOB: pour un dictionnaire de catégories avec
                    PUNCH=2
de type SASCATSQ: pour un dictionnaire de catégories avec
                    PUNCH=3
de type SASNORSQ: pour un dictionnaire de normes avec      PUN
                                                            CH=
                                                            3
de type SASCATNA: pour un dictionnaire de catégories avec  PUN
                                                            CH=
                                                            4
de type SASNORNA: pour un dictionnaire de normes avec PUNCH=4

```

Ces fichiers peuvent entrer directement dans une étape DATA sous la forme:

```

CMS COPYF INTRUSE8 SASCAT A = A (UNPACK;
DATA;
  SET SASCAT.INTRUSE8;
PROC PRINT LABEL;
PROC CONTENTS DATA=SASCAT.INTRUSE8;

```

faisant appel au fichier perforé "INTRUSE8 SASCAT" qui provient de la commande:

```
< ... CDWLOOK INTRUSE8 DIRE INTRUSE (SAS FM C PACK)>.
```

Remarquer la commande "CMS COPYF...(UNPACK;" dans l'encadré ci-dessus qui permet de décondenser le fichier "INTRUSE8 SASCAT" pour l'exécution du programme SAS.

Dans les fichiers SAS créés par le programme CDWLOOK, les noms des variables diffèrent pour chaque type de fichier (SASCAT, SASNOR, etc...). L'encadré ci-dessous détaille les noms des variables tels qu'ils peuvent être révélés par l'utilisation de la procédure PROC CONTENTS de SAS; cette même procédure permet aussi d'identifier les LABELS associés aux variables.

Codes pour dictionnaire de CATEGORIES:

Option SASCAT (PUNCH = 1)

```

SEG = Numéro de segment
WT  = Nombre total de mots dans le segment SEG
WD  = Nombre de mots différents dans le segment SEG
Fi  = Fréquence
RFi = SQRT Rate Fréquence
Di  = Densité
RDi = SQRT Rate Densité

```

Codes pour dictionnaire de CATEGORIES:

Option SASCATOB (PUNCH = 2)

CAT = Numéro de catégorie
WTi = Nombre total de mots dans le segment "i"
WDi = Nombre de mots différents dans le segment "i"
Fi = Fréquence
RFi = SQRT Rate Fréquence
Di = Densité
RDi = SQRT Rate Densité

Codes pour dictionnaire de NORMES:

Option SASNOR (PUNCH = 1 ou 2)

SEG = Numéro du segment
WT = Nombre total de mots dans le segment SEG
WD = Nombre de mots différents dans le segment SEG
TAT = Nombre total de mots assignés dans le segment SEG
TAD = Nombre de mots assignés différents dans le segment SEG
NORF = Somme des produits "norme x fréquence"
AF = Average Score Fréquence
NORD = Somme des produits "norme x densité"
AD = Average Score Densité

Codes pour dictionnaires de CATEGORIES:

Option SASCATSQ (PUNCH = 3)

SEG = Numéro du segment
SEQUENCE = Numéro de séquence du mot
CAT = Numéro de la catégorie

Codes pour dictionnaires de NORMES:

Option SASNORSQ (PUNCH = 3)

SEG = Numéro du segment
SEQUENCE = Numéro de séquence du mot
CAT = Numéro de la catégorie

Codes pour dictionnaires de CATEGORIES:

Option SASCATNA (PUNCH = 4)

WORD = Mot
SEG = Numéro du segment
SEQUENCE = Numéro de séquence du mot
CAT = Numéro de la catégorie

Codes pour dictionnaires de NORMES:

Option SASNORNA (PUNCH = 4)	
WORD	= Mot
SEG	= Numéro du segment
SEQUENCE	= Numéro de séquence du mot
CAT	= Numéro de la catégorie

M. Remarques

Pour illustrer ces informations, voici le déroulement de la création d'un fichier de type "SASNOR" depuis le fichier de type "WORDS" jusqu'à l'illustration graphique des valeurs de SQRT RATE FREQ.

A partir du fichier "JUAREZ EXEC" contenant (les guillemets font partie du fichier de type "EXEC"):

```
/* */
TRACE "C"
" ... CSCUT JUAREZ JUAREZ (NOSEND)"
IF RC > 8 THEN EXIT RC
" ... CRWSTRIP JUAREZ FRSTRP01 JUAREZ (NOSEND)"
IF RC > 8 THEN EXIT RC
" ... CDWLOOK JUAREZ UCLEM JUAREZ (SAS PACK"
EXIT RC
```

un fichier "JUAREZ SASNOR" est créé, qui est utilisé dans un fichier SAS tel que celui-ci:

```
OPTIONS LINESIZE=70 NONOTES;
CMS COPYF JUAREZ SASNOR A = A (UNPACK;
DATA;
    SET SASNOR.JUAREZ;
    SEGQ= SEG* SEG; SEGC= SEGQ* SEG;
PROC STEPWISE;
    MODEL AF = SEG SEGQ SEGC;
PROC REG;
    MODEL AF=SEG; ID SEG; OUTPUT OUT=JU P=P;
PROC PLOT;
    PLOT P * SEG='P' AF * SEG='*'/OVERLAY;
PROC CONTENTS DATA=SASNOR.JUAREZ;
```

qui, une fois exécuté, donne le résultat imprimé suivant:

```
1 options linesize=67 nonotes;
2 cms copyf juarez sasnor a = a (unpack;
3 data;
4     set sasnor.Juarez;
5     segq= seg* seg; segc= segq* seg;
6 proc stepwise;
7     model af = seg  segq  segc;
```

sas

12:12 wednesday, june 22, 1988

```

stepwise regression procedure for dependent variable af
note: slentry and slstay have been set to
      .15 For the stepwise technique.
Step 1      variable seg entered          r square = 0.43186522
                                         C(p) =      0.02926278
          Df      sum of squares      mean square      f      prob>f
regression      1      17.75473863      17.75473863      6.84      0.0280
Error           9      23.35702010      2.59522446
Total          10      41.11175873
          B value      std error      type iii ss      f      prob>f
intercept      33.68801818
Seg             0.40175455      0.15359997      17.75473863      6.84      0.0280
Bounds on condition number:          1,          1

```

```

-----
no other variables met the 0.1500 Significance level for entry
summary of stepwise regression procedure for dependent variable af

```

```

          variable      number      partial      model
step  entered  removed      in      r**2      r**2      c(p)
  1      seg          1          1      0.4319      0.4319      0.02926
          Variable
step  entered  removed      f      prob>f      label
  1      seg          1      6.8413      0.0280      Seg_juarez

```

```

8 proc reg;
9      model af=seg; id seg; output out=ju p=p;

```

```

          sas
          12:12 wednesday, june 22, 1988

```

```

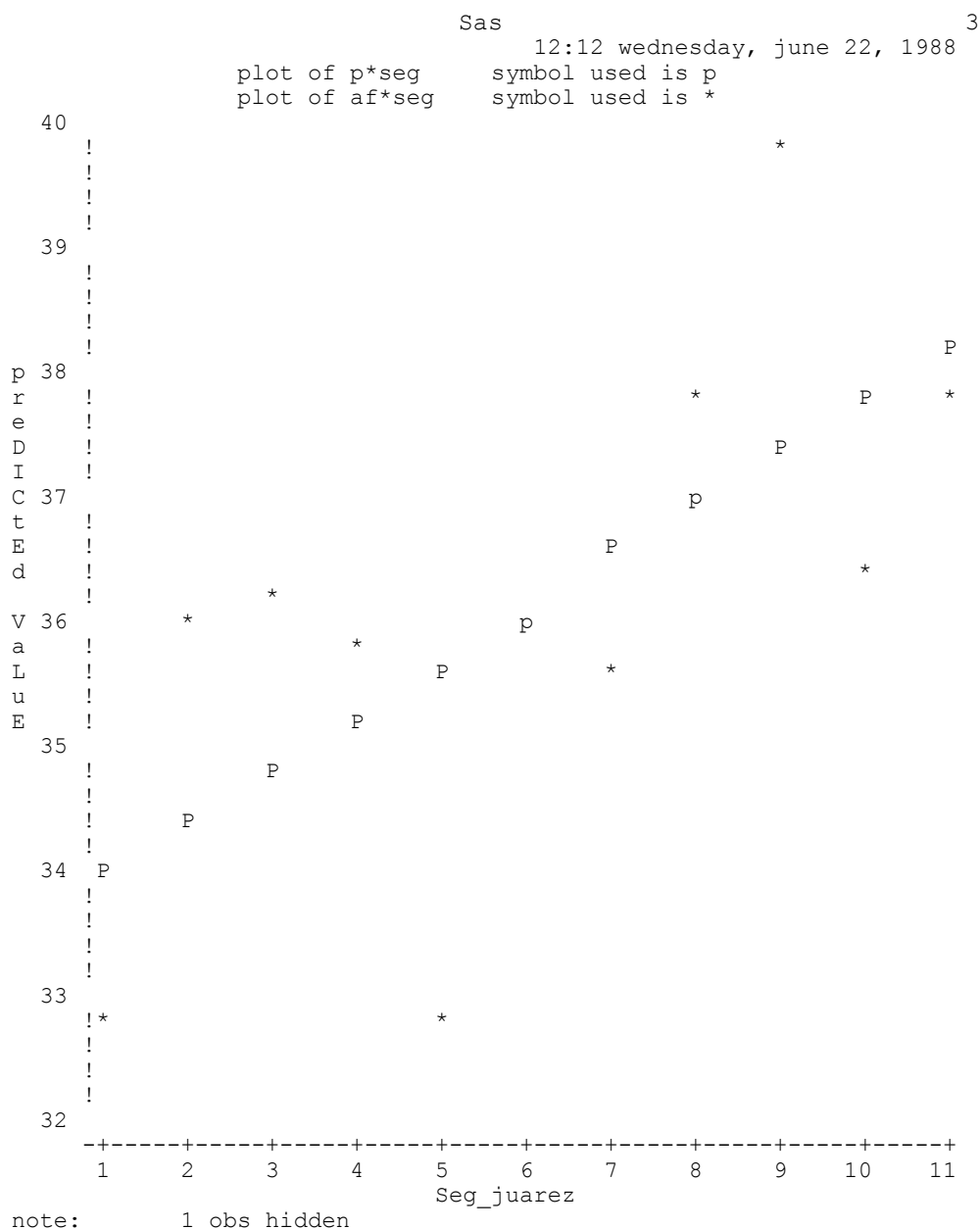
dep variable: af      af_uclem
          analysis of variance
          sum of      mean
source      df      squares      square      f value      prob>f
model       1      17.75473863      17.75473863      6.841      0.0280
Error       9      23.35702010      2.59522446
C total     10      41.11175873
          Root mse      1.61097      R-square      0.4319
          Dep mean      36.09855      Adj r-sq      0.3687
          C.V.          4.462701
          Parameter estimates
          parameter      standard      t for h0:
variable      df      estimate      error      parameter=0
intercep     1      33.68801818      1.04176566      32.337
Seg          1      0.40175455      0.15359997      2.616
          Variable
          prob > !T!      Label
intercep     1      0.0001      Intercept
seg          1      0.0280      Seg_juarez

```

```

10 proc plot;
11      plot p * seg='p' af * seg='*'/overlay;

```



Arrivé à ce point d'une analyse, il peut être éclairant de voir dans le texte en quoi, par exemple, les points 1 et 9 du graphique qui précède, diffèrent quant à leur degré d'émotionnalité. Ce graphique représente en effet le déroulement du taux d'émotionnalité dans la nouvelle "Histoire de Rosendo Juarez". Le point 1 (segment 1) étant le moins saturé en émotionnalité, et le point 9 (segment 9) étant le plus saturé, la question est de savoir quels sont, dans leur contexte, les mots qui rendent compte de cette différence. Les lignes qui suivent ont été produites par l'option "PRINTNA = 1" et ses options satellites. Elles permettent de suivre comment le dictionnaire UCLEM assigne certains mots du texte à certaines catégories, d'abord pour le segment 1, et ensuite pour le segment 9 (extraits).

" HISTOIRE DE ROSENDO JUAREZ " TIRE DE J.L. BORGES ,IN "LE RAPPORT

DE BRODIE", PP. 39-48.
 "UCLEM " NORMES D'EMOTIONS : AXE D'EMOTIONNALITE

WORDS TAGGED IN CONTEXT

TAG	SEG	SEQUENCE	WORDS & CATEGORIES
1	1	1	il devait e3tre onze <u>heures</u> (24) du <u>soir</u> (45)
		1	1 2 3 4 5 6 7
	1	8	j eltais entrel dans le bistrot qui est
		8	8 9 0 1 2 3 4 5
3	1	16	aujourd'hui un (21) bar à l <u>angle</u> (14) des
		16	6 7 8 9 0 1 2
5	1	23	<u>rues</u> (22) bolivar et venezuela l <u>homme</u> (43)
		23	3 4 5 6 7 8
7	1	29	dans son <u>coin</u> (22) me fit <u>signe</u> (28) d
		29	9 0 1 2 3 4 5
9	1	36	<u>approcher</u> (29) une <u>autoritel</u> (41)
		36	6 7 8
11	1	39	<u>incontestable</u> (23) devait elmaner de sa
		39	9 0 1 2 3
	1	44	personne car je lui obelis aussito3t il eltait
		44	4 5 6 7 8 9 0 1
12	1	52	assis à l une des <u>petites</u> (31) tables j eus
		52	2 3 4 5 6 7 8 9 0
13	1	61	de faclon inexplicable l <u>impression</u> (46) qu il
		61	1 2 3 4 5 6 7
14	1	68	y avait un (21) long <u>moment</u> (31) qu il eltait
		68	8 9 0 1 2 3 4 5
16	1	76	là sans <u>bouger</u> (29) devant son <u>verre</u> (47)
		76	6 7 8 9 0 1
			
357	9	1770	À relgler c est tout alors tu vas te <u>creler</u>
		1770	0 1 2 3 4 5 6 7 8 9
358	9	1779	(59) des ennuis pour un (21) inconnu et pour
		1779	0 1 2 3 4 5 6
359	9	1787	une <u>femme</u> (57) qui ne t <u>aime</u> (66) plus il ne
		1787	7 8 9 0 1 2 3 4 5
361	9	1796	voulut pas m <u>elcouter</u> (40) et s en fut le
		1796	6 7 8 9 0 1 2 3 4
	9	1805	lendemain on apprit qu il avait provoquel
		1805	5 6 7 8 9 0 1
362	9	1812	rufino dans un (21) bistrot de moroln et que
		1812	2 3 4 5 6 7 8 9
363	9	1820	rufino l avait <u>tuel</u> (59) il eltait allel à la
		1820	0 1 2 3 4 5 6 7 8
364	9	1829	mort et il s eltait fait <u>tuer</u> (59) comme il
		1829	9 0 1 2 3 4 5 6 7
365	9	1838	convient entre <u>hommes</u> (43) d <u>elgal</u> (30) à
		1838	8 9 0 1 2 3

Les indications TAG, SEG, et SEQUENCE indiquent, comme on peut le deviner, qu'il s'agit de la séquence des mots assignés par le dictionnaire, ensuite du numéro de segment, et enfin du numéro de séquence des mots dans le texte. Les numéros indiqués sous SEQUENCE désignent le premier mot de la ligne; le chiffre d'unité est repris en dessous de chacun des mots de la rubrique WORDS & CATEGORIES; les mots assignés sont soulignés et suivis du ou des numéros de catégories correspondants.

*CDWL281E	invalid category
*CDWL282E	duplicate category
*CDWL283E	category undefined in dictionary
*CDWL287E	option "NARR = 1" not provided to "CSCUT" - options "PRINTNA" & "PUNCH" (3,4) set to zero
*CDWL289E	options xxxxxxxx -- try to set xxxxxxxx to zero

XI. Le programme CFCHECK

A. But du programme

Le but du programme CFCHECK est de créer, à partir d'un fichier de type "INFIND", un fichier-système de type "FINDS" permettant de rechercher l'adresse exacte de toutes les occurrences d'un ensemble de mots. On a notamment en vue de pouvoir transformer les occurrences d'un mot infléchi d'une façon ou d'une autre selon le contexte où il se trouve, par exemple FERMES en FERMER (verbe) ou en FERME-J (adjectif).

B. Exécution

1. Appel du programme

```
... CFCHECK fn1 fn2 [fn3...]
```

a. où fn1 est le nom d'un fichier de filetype "CFCHECK" qui contient les paramètres et options. Ce fichier "fn1 CFCHECK" donnera son nom "fn1" au fichier de type "FINDS" qui sera normalement créé en sortie.

b. où fn2, fn3, etc... sont les noms des fichiers partiels de filetype "INFIND" qui contiennent tous les ou une partie des mots à rechercher (cfr. Chapitre IV "Les fichiers de données du système PROTAN et leur emploi" et Chapitre XXXII. "Annexes: Condensé de la syntaxe du fichier de données INFIND").

2. Entrées/sorties: Comme première entrée, CFCHECK demande un fichier "fn1 CFCHECK" contenant les options. Comme deuxième entrée, CFCHECK demande le ou les fichiers (jusque 99 fichiers partiels) de type "INFIND", composés des mots dont on va entreprendre la recherche d'adresse. Ce ou ces fichiers de type "INFIND" peuvent éventuellement avoir été créés par le programme CWKWOC (option "PUNCH = 4").

En sortie, CFCHECK crée un seul fichier-système de type "FINDS" (et de nom "fn1") pour les programmes CFLISTA et CFWKWIC.

3. Paramètres et options:

LIST1 = Par défaut ("LIST1 = 0"), il n'y a pas impression de la phase 1 de l'exécution. Autrement ("LIST1 = 1"), il y a impression.

La phase 1 de l'exécution d'un fichier de mots est la phase durant laquelle le programme traite le fichier tel qu'il a été construit, les mots y étant rangés dans leur ordre original. A la phase 1 succède une phase 2 où le fichier de mots est trié en ordre alphabétique avec suppression simultanée des mots pour lesquels une erreur a été signalée lors de la phase 1.

LIST2 = Par défaut ("LIST2 = 0"), il n'y a pas impression de la phase 2. "LIST2 = 1" déclenche cette impression.

FIND = "FIND = 1", le défaut, permet la constitution d'un fichier-système de type "FINDS" utilisable par les programmes CFLISTA et

CFWKWIC. "FIND = 0" inhibe la création de ce fichier.

C. Exemples

1. Exemples de procédure d'appel

a. " ... CFCHECK ISIDA2 ISIDA"

Le fichier "ISIDA2 CFCHECK" contient les options du programme qui créera le fichier "ISIDA2 FINDS" à partir du fichier de mots "ISIDA INFIND".

b. Exemple de fichier de paramètres et options

(1) Contenu du fichier "ISIDA2 CFCHECK"

(entre < et >):

<LIST1 = 1, LIST2 = 1;>

On demande l'impression des phases 1 et 2, l'option "FIND = 1" étant prise par défaut.

D. Préparation des données

Ce programme ne demande aucune préparation particulière. Le programme CFCHECK et tout ce qui s'y rattache, c'est-à-dire les fichiers de type INFIND et FINDS, et les programmes CFWKWIC et CWEDIT, ont pour objectif de traiter l'ambivalence du langage. Il y a à ce sujet une stratégie d'analyse qui peut être efficace et qui consiste à constituer peu à peu une liste de mots courants mais aussi ambigus qui se retrouvent dans la majorité des textes. Des mots comme "FERME", "PORTE", et d'autres pourraient se retrouver dans un fichier "permanent" de mots à désambigüiser. A ce fichier permanent s'ajouteraient alors un fichier des mots ambigus limités au texte dont on réalise l'analyse.

On retiendra que tous les types d'items peuvent être spécifiés dans un fichier INFIND, outre des mots, par exemple un marqueur ou un signe de ponctuation.

E. Autres informations

Aucune.

F. Remarques

Aucune.

G. Messages propres au programme CFCHECK

*CFCH651E	duplicate words/roots = nnnnnn
*CFCH652U	user's findlist length > 99999
*CFCH653E	inaccurate alignment
*CFCH654E	null root
*CFCH655E	word/root syntax error
*CFCH662E	invalid exclude code
*CFCH663E	words/roots purged = nnnnnn
*CFCH664E	duplicate root - latest one in use
*CFCH665E	duplicate word - first one in use

XII. Le programme CFLISTA

A. But du programme

CFLISTA permet l'impression du fichier de type "FINDS" créé par le programme CFCHECK.

B. Exécution

1. Appel du programme:

```
... CFLISTA fn1 fn2
```

a. où fn1 est le fichier de paramètres et options,

b. où fn2 est le fichier de filetype "FINDS", créé par CFCHECK, qui contient les mots ou racines à rechercher.

2. Entrées/sorties: CFLISTA ne crée aucun fichier; ce programme ne fait qu'imprimer la liste des mots du fichier de type "FINDS".

3. Paramètres et options:

SFINDS = Cette option est nécessaire pour l'utilisation de PROTAN en PC, pour fournir au programme le nom (et le chemin éventuellement) du fichier FINDS. Par exemple, *SFINDS* = 'd:....CAMUSfch.fds'.

C. Exemples

1. Exemple de procédure d'appel

a. " ... CFLISTA ISIDA ISIDA".

On demande l'impression du contenu du fichier de mots "ISIDA FINDS" issu de l'exécution du programme CFCHECK.

D. Préparation des données

Aucune.

E. Autres informations

Aucune.

F. Remarques

Aucune.

G. Messages propres au programme CFLISTA

```
*CFLA681E duplicate root - latest one in use
*CFLA682E duplicate word - first one in use
```


XIII. Le programme CFWKWIC

A. But du programme

Le but du programme CFWKWIC est de mettre les mots sélectionnés par l'analyste (par l'intermédiaire du programme CFCHECK) dans leur contexte de départ tout en créant parallèlement un fichier "PUNCH" permettant à l'analyste de modifier ces mots en fonction du sens qu'ils prennent dans leur contexte.

B. Exécution

1. Appel du programme

```
... CFWKWIC fn1 fn2 fn3
```

a. où fn1 est le nom d'entrée (filename) d'un fichier de type CFWKWIC; en sortie, ce nom devient celui d'un fichier de filetype "CWEDIT";

b. où fn2 est le nom du fichier-système de filetype "FINDS", créé par CFCHECK.

c. où fn3 est le nom du fichier-système de filetype "WORDS" (qui peut avoir été créé par les programmes "CSCUT" --avec option "NARR=1"--, "CRWSTRIP", ou même "CWEDIT").

2. Entrées/sorties: Le programme CFWKWIC demande trois fichiers **d'entrée**. Il demande d'abord un fichier "fn1 CFWKWIC" qui contienne les paramètres et options. Il demande ensuite un fichier "fn2 FINDS"; celui-ci, créé par CFCHECK, est un fichier-système contenant les mots ou racines dont l'analyste souhaite localiser les occurrences et les adresses. Enfin, le programme demande un fichier-système "fn3 WORDS" qui est le corpus créé par CSCUT, CRWSTRIP, ou même une exécution antérieure de CWEDIT; on retiendra que l'option "NARR = 1" devra avoir été fournie au programme CSCUT qui est à l'origine du fichier de filetype "WORDS" requis ici comme troisième fichier d'entrée.

En **sortie**, CFWKWIC crée le fichier "fn1 CWEDIT" ("fn1" étant le nom du fichier de type "CFWKWIC") qui contient les paramètres et options déjà toutes prêtes pour le programme suivant CWEDIT.

3. Paramètres et options:

STAT = Par défaut, "STAT = 0", il n'y a pas impression des statistiques de tri qui accompagnent les fichiers de type "WORDS". "STAT = 1" donne les statistiques du nombre de mots différents et du nombre total de mots, par segment et pour le total.

PRINT = Les options "PRINT = 1" et "PRINT = 2" ont pour effet que les mots (ou racines) que l'analyste souhaite éventuellement modifier sont à présent imprimés entourés, de part et d'autre, des mots qui les précèdent (par défaut, 10 items, voir option ANTE =) et des mots qui les suivent (par défaut, 5 items, voir option POST =), avec

en outre des indications sur la position exacte où se trouve chaque occurrence de ces mots et items dans le texte. L'option "PRINT = 1", l'option par défaut, imprime les occurrences de chaque type en ordre alphabétique, tandis que l'option "PRINT = 2" les imprime dans l'ordre séquentiel original du texte. L'option "PRINT = 0" inhébe toute impression.

PUNCH = Cette option complète la précédente. Connaissant le contexte de chaque occurrence de chaque mot recherché et son numéro de séquence, on peut procéder à présent à la modification éventuelle de type "ancien mot ... nouveau mot" de chaque occurrence des mots recherchés. En effet, à chaque ligne de contexte imprimé par les options "PRINT = 1" ou "PRINT = 2" correspondra une ligne du fichier "PUNCH". Ce que le programme CFWKWIC crée avec ce fichier PUNCH est en fait le fichier de filetype "CWEDIT" avec ses options et paramètres; le programme CWEDIT est l'étape qui suit normalement l'exécution de CFWKWIC.

Comme dans l'option PRINT précédente, on dispose d'une option "PUNCH = 1", le défaut, donnant les occurrences de chaque type de mot recherché en ordre alphabétique; et on dispose d'une option "PUNCH = 2" donnant ces mêmes occurrences dans l'ordre séquentiel original du texte. "PUNCH = 0" inhébe toute création de fichier PUNCH. Notons que ce fichier PUNCH ne fournit que les occurrences des mots recherchés sans leur contexte.

ANTE = L'option ANTE, ainsi que la suivante, POST, permet de contrôler le nombre de mots ou items qui précèdent (ANTE =) et qui suivent (POST =) les mots-clés. Par défaut, ANTE = 10.

POST = La valeur par défaut de cette option est (POST = 5).

C. Exemples

1. Exemple de procédure d'appel

a. " ... CFWKWIC camus3 camus2 camus1" (en VM).

Le fichier "camus3 CFWKWIC" contient les options du programme qui créera à son tour le fichier CWEDIT qui, lui, portera le nom "camus3 CWEDIT". Le fichier "camus2 FINDS" est le fichier--système des mots à rechercher; ce fichier aura été créé antérieu-

rement par le programme CFCHECK¹³. Et le fichier "camus1 WORDS" est le corpus, également sous forme de fichier-système (il s'agit ici des deux premiers chapitres de La Chute de Camus).

b. En DOS, la procédure d'appel aurait consisté à exécuter le fichier "CFWKWIC1.BAT" contenant:

```
call protan.bat cfkwkic camusfwk.lis camusfwk.fwk
if errorlevel 12 goto exit
:exit
```

où "camusfwk.lis" définit le nom du listing produit par le programme dont les paramètres et options figurent dans "camusfwk.fwk" (voir ci-après).

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "camus3 CFWKWIC" (ou "camusfwk.fwk" (entre < et >)¹⁴:

```
<
stat = 1, print=1, punch=1, ante = 10, post = 5,
swords = 'camusrws.wds', sfinds='camusfch.fds',
spunch = 'camuswed.wed';
>
```

On souhaite que l'impression et la perforation des occurrences des mots recherchés soient faites en ordre alphabétique toutes les deux. Il y aura 10 items de contexte devant les mots-clés, et 5 après. Le fichier de type "WORDS" est défini, ainsi que le fichier de type "FINDS" et les paramètres de CWEDIT (qui est le programme qui suit logiquement CFWKWIC).

D. Préparation des données

Aucune.

E. Autres informations

Les fichiers de départ étant (voir exemple au point C.1 et C.2 ci-dessus):

- "camus3 CFWKWIC", les options,
 - "camus2 FINDS", le fichier-système créé à partir des fichiers "camus2 CFCHECK" et "camus INFIND" (contenant les mots à rechercher: "ENTRE", "MANQUE", et "PARTI"),
- et
- "camus1 WORDS", le corpus,
- l'exécution de la commande:

```
"EXEC CFWKWIC camus3 camus2 camus1"
```

¹³ "... CFCHECK camus2 camus"

- où "camus2 CFCHECK" est le fichier de paramètres et options qui donnera son nom au fichier "camus2 FINDS",
- et où "camus INFIND" est le fichier contenant les mots-clés.

¹⁴ Les définitions de fichiers "swords", "sfinds", et "spunch" ne sont utiles que dans un environnement DOS.

donne un résultat imprimé et un résultat perforé (en fait, un fichier de type "CWEDIT"). Le résultat imprimé se présente comme suit:

```
**          PROTAN SYSTEM          UCL/PSP AT LOUVAIN-LA-NEUVE
PAGE:      3  **
**          PROGRAM: CFWKWIC        DATE: Apr27 93    TIME: 21:46:45
**
** CAMUS
**
"WORDS" FILE DEFINITIONS _____
"WORDS" FILE IDENTIFICATION (USER-DEFINED)..... CAMUS1
"WORDS" FILE CREATION DATE..... Apr27 93
"WORDS" FILE CREATION TIME..... 21:46:05
USER'S COMMENT.

"WORDS" FILE PROCESSING LEVEL..... 1

PROCESSING OPTIONS PROVIDED TO "CSCUT"

LBRKB='I';
BRKM=1;
CTRL=53;
MODULO=0;
NARR=1;
REGEN=0;
RESIDUE='KEEP';
SEGT='S';
SENT='.!;?:';

NUMBER OF SEGMENTS GENERATED : 4

GENERAL COMMENTS (HISTORY OF "SOURCE" & "WORDS" FILES AND OTHER
COMMENTS)

PROC BY CRWSTRIP "CAMUS1 " (Apr27 93 21:46:05) "FRSTRP01" (Nov01
91 09:47:55)
PROC BY CSCUT    "CAMUS1 " (Apr27 93 21:45:15)

CAMUS LA CHUTE (1)

TOTAL NUMBER OF ITEMS SELECTED : 2
**          PROTAN SYSTEM          UCL/PSP AT LOUVAIN-LA-NEUVE
PAGE:      4  **
**          PROGRAM: CFWKWIC        DATE: Apr27 93    TIME: 21:46:50
**
** CAMUS
**
** CAMUS1
**

                                MATCHED WORDS IN CONTEXT

segment first line(s): 10 items before matched word
keyword  sequence      matched word
                                last line(s): 5 items after matched word
-----
AU MOINS , QUE QUELQUE CHOSE NE TOURNER NON ROND
      1      467      ENTRE      ENTRE
                                _____
LES HOMME . CETTE DISPOSITION
-----
PAR MYTHOMANIE OU PAR BE3TISE . PAR EXCE2S OU PAR
```

Le résultat perforé, qui est en fait un fichier de type "CWEDIT", se présente de manière telle qu'il n'y a plus qu'à modifier, si on le souhaite, les occurrences des divers mots-clés. Par exemple, connaissant le contexte des mots "entre" et "manque", décider de laisser "entre" inchangé (car, à la séquence 467, il s'agit d'une préposition), et décider de transformer "manque" en "manquer" (car on considère, pour l'une ou l'autre raison, qu'on peut ou qu'on doit transformer ce mot en sa forme verbale à la séquence 1053).

```
FID='CAMUS      ',
FDATE='Apr27 93',
FTIME='21:36:59',
FCOM='
';
      467  ENTRE                ENTRE
      1053  MANQUE              MANQUE
```

F. Remarques

Aucune.

G. Messages propres au programme CFWKWIC

```
*CFWK701S  option "NARR = 1" not provided to "CSCUT" - "WORDS"
           file cannot be processed
*CFWK702S  no item has been selected in "WORDS" file
```

XIV. Le programme CRCHECK

H. But du programme

Le but du programme CRCHECK est de créer le dictionnaire de formes nominales (cfr. Chapitre XXXII. Annexes: Les dictionnaires de formes nominales et les catégories) contenant d'une part les formes entières des mots, les lexies si on veut, et d'autre part, les formes nominales de ces lexies. Secondairement, le programme CRCHECK effectue un test de comparaison entre lexies et formes nominales afin de détecter la présence de "transformations en boucle".

I. Exécution

1. Appel du programme

```
... CRCHECK fn1 fn2 [fn3...]
```

a. où fn1 est le nom d'un fichier dont le filetype est CRCHECK, contenant paramètres et options. Ce fichier "fn1 CRCHECK" donnera son nom "fn1" au fichier de type "ROOTS" qui sera constitué en sortie.

b. où fn2, fn3, etc... sont les noms des fichiers partiels de filetype "INROOT" qui contiennent chacun une partie d'un dictionnaire de formes nominales (cfr. Chapitre IV. "Les fichiers de données du système PROTAN et leur emploi" et Chapitre XXXII. Annexes: Condensé de la syntaxe du fichier de données INROOT).

2. Entrées/sorties: Comme première entrée, CRCHECK demande un fichier "fn1 CRCHECK" contenant les options. CRCHECK demande une deuxième entrée, à savoir le ou les fichiers (jusque 99 fichiers partiels) de type "INROOT", qui constituent le dictionnaire de formes nominales.

En sortie, CRCHECK crée un fichier-système de type "ROOTS" (et de nom "fn1") pour les programmes CRLISTA et CRWSTRIP.

3. Paramètres et options:

LIST1 = Par défaut ("LIST1 = 0"), il n'y a pas impression de la phase 1 de l'exécution. Autrement ("LIST1 = 1"), il y a impression.

La phase 1 de l'exécution d'un dictionnaire est la phase où le programme traite le dictionnaire tel qu'il a été construit, les mots y étant rangés dans leur ordre original. Cette phase 1 est suivie d'une phase 2 où alors les mots du dictionnaire sont rangés dans un ordre alphabétique d'où sont effacés tous les mots à propos desquels une erreur a été signalée lors de la phase 1.

LIST2 = Par défaut ("LIST2 = 0"), il n'y a pas impression de la phase 2. "LIST2 = 1" déclenche cette impression.

CROSS = Un dictionnaire de formes nominales constitue nécessairement un volume d'entrées important. Le dictionnaire de formes nominales "FRSTRP01", de plus de 9.000 entrées, en est une illustration. Au cours de la construction d'un dictionnaire, il arrive que se modifient les critères sur base desquels on décide de réduire les formes différentes des mots. Par exemple, dans un premier temps, on jugera utile de transformer les occurrences de "autrement" en la forme réduite "autre", alors que, dans un second temps, on décidera, par oubli du critère antérieur, de transformer "autre" en la forme "autrui". Un second passage par le dictionnaire de formes nominales provoquerait une altération importante du fichier de type "WORDS": en effet, les formes "autrement", déjà transformées en "autre" lors du premier passage, seraient transformées à présent en "autrui".

Par défaut ("CROSS = 0"), ce test n'est pas effectué. L'option "CROSS = 1" permet de repérer les cas de réductions croisées: un message d'avertissement est produit ("WARNING" sur la liste). L'option "CROSS = 2" repère les mêmes cas, et repère en outre les cas de transformation qui n'aboutiraient pas à une altération du mot. Par exemple, "chante." et "chant." transformés tous les deux en "chanter".

ROOT = "ROOT = 1", le défaut, entraîne la création du fichier-système de type "ROOTS" --en fait, le dictionnaire de formes réduites utilisable par les programmes CRLISTA et CRWSTRIP--. "ROOT = 0" inhibe la création de ce fichier-système.

J. Exemples

1. Exemples de procédure d'appel

a. " ... CRCHECK FRSTRP01 FRSTRPA FRSTRPB
FRSTRPC"

Dans cet exemple, le fichier "FRSTRP01 CRCHECK" contient les paramètres et les options du programme. Le nom FRSTRP01 deviendra le nom du fichier de type "ROOTS" qui sera constitué. Par ailleurs, les fichiers "FRSTRPA INROOT", "FRSTRPB INROOT", et "FRSTRPC INROOT" contiennent chacun une partie des données du dictionnaire ainsi divisé en 3 parts.

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "FRSTRP01 CRCHECK" (entre
< et >):

<LIST1 = 1, LIST2 = 1, CROSS = 1;>

On demande l'impression des phases 1 et 2 ainsi que le test des réductions croisées. Le fichier "FRSTRP01 ROOTS" est créé par défaut.

K. Préparation des données

La préparation d'un fichier de données de type INROOT peut être assez longue, même si on ne vise qu'à compléter un dictionnaire existant. Un moyen efficace de préparer un tel dictionnaire de formes nominales consiste à le générer de manière semi-automatique par le programme CWKWOC (avec l'option "PUNCH = 2"). Avec ce programme, les seuls deux mots-clés 'AIT~' et 'AIENT~' (~ = caractère blanc) permettent par exemple d'obtenir un fichier contenant, dans les colonnes 1 à 20 --redoublées en colonnes 24 à 43-- les formes conjuguées de la troisième personne, singulier et pluriel, de l'imparfait et du conditionnel de tous les verbes contenus dans un fichier de type "WORDS". Il suffit alors de remplacer les formes conjuguées, figurant dans les colonnes 24 à 43, par l'infinitif du verbe pour obtenir un fichier de type "INROOT". (Voir aussi Chapitre IV. Les fichiers de données du système PROTAN et leur emploi).

On retiendra que tous les types d'items peuvent être spécifiés dans un fichier INROOT, outre des mots, soit par exemple un marqueur ou un signe de ponctuation.

L. Autres informations

Un fichier de type "INROOT" est normalement composé de deux séries de mots, en regard l'une de l'autre, le mot à réduire à gauche et le même mot réduit à droite. Par exemple, la lexie "heureuses" et sa forme nominale "heureux". Mais un programme de réduction est aussi un programme d'édition en quelque sorte. Dans cet ordre d'idées, on peut imaginer de construire plusieurs dictionnaires de type "ROOTS" et de procéder à plusieurs exécutions successives du programme CRWSTRIP sur le même texte à l'aide de différents dictionnaires de type "ROOTS", chaque dictionnaire effectuant un type d'édition particulier. Par exemple, s'il apparaît que toutes les occurrences du mot "TERRE", dans une série de textes, doivent être changées en "\$TERRE" pour l'une ou l'autre raison, alors qu'on est déjà loin dans les étapes de l'analyse, on pourra, à peu de frais, inclure la paire "TERRE ...\$TERRE" dans un fichier de "INROOT" et procéder à cette modification systématique dans le texte.

Un autre exemple d'édition par réduction est suggéré par la possibilité de transformer, dans le fichier INROOT, un mot ou une racine en rien. Ceci permet alors de rassembler dans un fichier INROOT séparé, toutes les lexies qu'on veut voir disparaître d'un texte, par exemple les articles, pronoms, prépositions, etc., soit la majorité des mots-outils de la langue.

M. Remarques

Aucune.

N. Messages propres au programme CRCHECK

*CRCH301E	duplicate words/roots = nnnnnn
*CRCH302W	number of cross-equivalents found (warning) = nnnnnn
*CRCH304U	user's striplist length > 99999
*CRCH306E	inaccurate alignment
*CRCH307E	null word/root at left side
*CRCH308E	null root at right side
*CRCH309E	word/root syntax error
*CRCH310E	impossible word to root conversion
*CRCH311E	invalid substitution
*CRCH312W	cross-equivalent generator
*CRCH313E	words/roots purged = nnnnnn
*CRCH314E	duplicate root - latest one in use
*CRCH315E	duplicate word - first one in use
*CRCH318I	number of cross-equivalents found (inform.) = nnnnnn

XV. Le programme CRLISTA

A. But du programme

CRLISTA imprime le dictionnaire de formes nominales créé par CRCHECK, soit en ordre alphabétique sur les lexies, c'est-à-dire sur la partie gauche du dictionnaire, constituée des formes entières des mots, soit en ordre alphabétique sur les unités minimales, c'est-à-dire la partie droite du dictionnaire, constituée de la forme nominale des mots ou du radical de ces mots (ou laissée vide si on décide de procéder par suppression de mots).

B. Exécution

1. Appel du programme:

```
... CRLISTA fn1 fn2
```

a. où fn1 est le nom d'un fichier de filetype "CRLISTA" qui contient les paramètres et les options.

b. où fn2 est le nom d'un fichier de filetype "ROOTS", créé par CRCHECK, qui constitue le dictionnaire des formes nominales.

2. Entrées/sorties: CRLISTA ne crée aucun fichier; ce programme a seulement besoin, en entrée, du fichier de type "ROOTS" ainsi que du fichier d'options.

3. Paramètres et options:

SORTT = Par défaut ("SORTT = 'OLD'"), le dictionnaire de formes nominales est imprimé en ordre alphabétique sur les formes entières et conjuguées des mots. "SORTT = 'NEW'" permet l'impression en ordre alphabétique sur les formes nominales.

C. Exemples

1. Exemple de procédure d'appel

a. " ... CRLISTA DOUZE FRSTRP01"

On demande l'impression du dictionnaire "FRSTRP01 ROOTS" selon les paramètres décrits dans le fichier "DOUZE CRLISTA".

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "DOUZE CRLISTA" (entre < et >):

<;>

On demande que les mots du dictionnaire soient rangés alphabétiquement sur la colonne de gauche, c'est-à-dire sur les formes entières des mots.

D. Préparation des données

Aucune.

E. Autres informations

Aucune.

F. Remarques

Aucune.

G. Messages propres au programme CRLISTA

*CRLA331S	invalid sort type
*CRLA332W	cross-equivalent generator
*CRLA333E	duplicate root - latest one in use
*CRLA334E	duplicate word - first one in use

XVI. Le programme CRWSTRIP

A. But du programme

Le but de CRWSTRIP est de réduire le plus grand nombre de formes différentes d'un même mot (féminin, pluriel, formes conjuguées) à une forme unique minimale, voire à une racine; CRWSTRIP permet également de supprimer des mots du corpus. On arrive à ce résultat par la construction de dictionnaires de formes nominales appropriés d'une part, et par l'exécution du programme CRWSTRIP sur un fichier-système de type "WORDS" (créé par les programmes CSCUT, CRWSTRIP, CWADD, ou CWEDIT) d'autre part.

Dans sa forme la plus simple, CRWSTRIP reçoit un fichier de type "WORDS" qu'il transforme --réduit-- à l'aide d'un dictionnaire de formes nominales pour créer un nouveau fichier-système de type "WORDS", utilisable par le programme CRWSTRIP lui-même, pour commencer¹⁵, et par les programmes CDWLOOK, CWKWOC, CFWKWIC, CWADD, CWEDIT, CWWCOL, CWORD, CWNEW, CWTALLY, et CWSELECT ensuite.

B. Exécution

1. Appel du programme

```
... CRWSTRIP fn1 fn2 fn3
```

a. où fn1 est le nom (filename) d'un fichier dont le type (filetype) est CRWSTRIP en entrée; ce même fn1 sera le nom, en sortie, d'un fichier dont le type sera "WORDS", "PUNCHW", ou "SASWORD";

b. où fn2 est le filename d'un fichier de filetype "ROOTS" --le dictionnaire de formes nominales créé par CRCHECK--;

c. où fn3 est le filename du fichier de filetype "WORDS", créé précédemment par CSCUT, CWEDIT, ou par une exécution antérieure de CRWSTRIP, et appelé ici en entrée par CRWSTRIP.

2. Entrées/sorties: En entrée, CRWSTRIP appelle trois fichiers qui sont respectivement le fichier "fn1 CRWSTRIP" (paramètres et options), le fichier "fn2 ROOTS" (le dictionnaire de formes nominales sous forme de fichier-système), et enfin le fichier "fn3 WORDS" (le corpus, transformé en un fichier-système de type "WORDS" par CSCUT, CWEDIT, ou par une exécution antérieure de CRWSTRIP).

En sortie, et par défaut d'ailleurs, CRWSTRIP crée un fichier-système de type "WORDS". Ce fichier "WORDS", lisible seulement par les programmes qui en ont besoin, contient notamment la liste alphabétique des mots différents contenus dans le ou les textes, avec leurs fréquences, et d'autres statistiques de tri. Les options "TABLE = 1" et "STAT = 1" permettent une impression lisible de cette liste et de ces statistiques de tri. L'option "PUNCH = 1" permet la perforation de cette liste sous forme de fichiers "PUNCHW" ou "SASWORD".

¹⁵ Si on utilise plusieurs dictionnaires différents de formes nominales.

3. Paramètres et options:

<i>TABLE</i> =	La table des mots triés en ordre alphabétique peut être imprimée par cette option "TABLE = 1". "TABLE = 0", le défaut, inhibe l'impression.
<i>STAT</i> =	Le défaut de cette option, "STAT = 1", imprime les statistiques de tri, c'est-à-dire le nombre de mots différents, le nombre total de mots, et le rapport type-occurrences --TTR--. Il y a même une double statistique de tri qui est présentée, respectivement pour le nombre de mots (occurrences et types) avant l'action de CRWSTRIP et après cette action. "STAT = 0" inhibe cette impression.
<i>ROOTT</i> =	"ROOTT = 1" permet de suivre les transformations qui ont été effectuées par le dictionnaire de formes nominales. Par exemple, on aura "HEUREUSEMENT" transformé en "HEUREUX" si une telle transformation est prévue dans le dictionnaire. "ROOTT = 0", le défaut, inhibe l'impression de la chaîne des transformations.
<i>PUNCH</i> =	"PUNCH = 1" perfore le contenu de la liste imprimée par le paramètre "TABLE = 1", en créant un fichier de type "PUNCHW" ou "SASWORD". (Voir les options des procédures d'appel). Le défaut, "PUNCH = 0", inhibe la création de ce fichier de type "PUNCHW" ou "SASWORD". Pour le format de perforation de ces fichiers de type "PUNCHW" ou "SASWORD", le lecteur voudra bien se reporter au point "Remarques" du programme CSCUT.
<i>WORD</i> =	Par défaut ("WORD = 1"), CRWSTRIP crée toujours un fichier-système de type "WORDS" qui sera utilisé par les programmes CDWLOOK, CWKWOC, CWWORD, CFWKWIC, CWADD, CWEDIT, CWNEW, CWWCOL, CWTALLY, CWSELECT, et par CRWSTRIP lui-même si on utilise différents dictionnaires de formes nominales. (Note: Ce fichier "WORDS" peut être utilisé condensé par les programmes mentionnés ci-dessus). "WORD = 0" inhibe la création de ce fichier "WORDS".

C. Exemples

1. Exemple de procédure d'appel

a. " ... CRWSTRIP DOUZE2 FRSTRP01 DOUZE1".

Un premier fichier "DOUZE2 CRWSTRIP" contient les paramètres et les options. Un deuxième fichier "FRSTRP01 ROOTS" est le dictionnaire de formes nominales qui va "réduire" le troisième fichier "DOUZE1 WORDS".

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "DOUZE2 CRWSTRIP" (entre < et >):

```
<TABLE = 0, STAT = 0, ROOTT = 1;>
```

On ne désire pas inspecter la table des mots triés ni les statistiques de tri (ni en entrée ni en sortie), mais on veut vérifier l'efficacité du dictionnaire de formes nominales sur le texte ("ROOTT = 1"). Il y aura création d'un fichier "DOUZE2 WORDS", le filename de ce fichier étant fixé par le filename du fichier de paramètres "DOUZE2".

D. Préparation des données

Aucune.

E. Autres informations

Bien que l'exécution de chacun des programmes du logiciel PROTAN soit rapide, l'ensemble d'une analyse, depuis le premier programme de vérification des données (CSCHECK) jusqu'aux programmes d'analyse catégorielle ou non-catégorielle, peut parfois être complexe: On peut avoir plusieurs corpus différents à analyser, chaque analyse générant ses propres fichiers-système. Il pourrait aussi arriver qu'à un moment, on ne sache plus si un corpus a ou n'a pas été soumis au programme CRWSTRIP. C'est que CRWSTRIP comme CSCUT génèrent en effet un fichier-système qui est toujours de filetype "WORDS". Si un tel fichier-système issu de CRWSTRIP porte en plus le même filename que celui du fichier "WORDS" créé par CSCUT, ce fichier "WORDS" plus ancien sera "écrasé" par le plus récent. Ceci n'est pas un problème en soi tant qu'on sait qu'il s'agit d'un fichier issu de CRWSTRIP.

Pour cette raison, à partir du moment où un fichier de type "WORDS" a été traité par CRWSTRIP, il porte mention de ce traitement. Ainsi, si plusieurs dictionnaires de formes nominales sont utilisés sur le même corpus, chaque exécution du programme CRWSTRIP garde la mémoire cumulative de ces traitements. Cette mémoire est signalée, dans le listing ou le fichier PUNCH, sous la rubrique "GENERAL COMMENTS", par le message :

```
"PROC BY CRWSTRIP"
```

et, sur la même ligne, l'indication de la date et de l'heure de ce traitement, elle-même suivie du nom du dictionnaire de formes nominales utilisé avec l'indication de la date et de l'heure de sa création.

F. Remarques

Aucune.

G. Messages propres au programme CRWSTRIP

*CRWS351S no word to include in output WORDS file

XVII. Le programme CSCHECK

A. But du programme

CSCHECK a pour but de vérifier si le texte présent entre les colonnes 1 à 70 et les codes de segmentation présents entre les colonnes 73 à 80 ont été entrés en accord avec la syntaxe imposée pour un fichier-source (décrite au Chapitre IV "Les fichiers de données du système PROTAN et leur emploi"). CSCHECK ne crée aucun fichier, sinon une liste où figurent divers messages d'informations, d'avertissements, ou d'erreurs plus ou moins graves. Ces messages concernent l'état d'un mot du texte ou d'un code de segmentation particulier.

B. Exécution

1. Entrées/sorties: Comme tous les programmes de PROTAN, CSCHECK demande un fichier de paramètres et options en entrée et crée un fichier listing en sortie. Le fichier de paramètres et options doit identifier les entrées et sorties, soit:

a. **Entrées:** un ou des fichiers SOURCE (partiels) contenant le ou les textes à analyser. Les noms de fichiers sont donnés via l'option SSOURCE (voir paramètres et options).

b. **Sortie(s):** En sortie, CSCHECK crée seulement une liste indiquant les éventuelles erreurs contenues dans le fichier "source".

2. Paramètres et options:

PRINT= Si PRINT=0 (la valeur par défaut), le texte n'est pas imprimé et seuls sont imprimés les messages générés par l'exécution de CSCHECK ainsi que les commentaires qui font partie du texte. Si PRINT=1, le texte et les messages sont imprimés.

SENT = Cette option définit les caractères séparateurs de phrases qui vont être reconnus par le système PROTAN. En l'absence d'autre indication, les cinq caractères suivants:

< ; . ! ? : >

sont les séparateurs de phrases qui fonctionnent par défaut dans le système; si on veut, on peut écrire le défaut de la façon suivante:

<SENT = ' ; . ! ? ; ' > ,

ou plus simplement:

<SENT = ' ' > .

On remarque que les caractères séparateurs de phrases sont entourés d'une apostrophe de part et d'autre et qu'aucun caractère blanc (~) n'est contenu entre les apostrophes.

Il est cependant possible d'imposer au système les caractères qu'on veut voir reconnus comme séparateurs de phrases. On peut ainsi compléter la liste des séparateurs de phrases qui fonctionnent déjà par défaut, ou la modifier, ou encore la remplacer par d'autres caractères qu'on choisit alors dans la liste

des caractères séparateurs de mots. Il y a deux exceptions pourtant:

- a. le caractère blanc (~) ne peut pas être utilisé comme séparateur de phrases, et
- b. l'apostrophe (') ne peut, elle, être utilisée qu'à condition d'être redoublée dans la description de l'option. (**Note:** ne pas confondre l'apostrophe redoublée avec le guillemet.)

On pourra donc avoir, par exemple,

```
<SENT = './'>
```

(les seuls séparateurs de phrases reconnus sont le point et la barre oblique), ou encore ceci,

```
<SENT = '!''. '>
```

(pour quelque raison, on souhaite que l'apostrophe notamment soit reconnue comme caractère séparateur de phrases dans un texte).

COMM = Si COMM = 0, les commentaires insérés par l'utilisateur dans le texte (lignes de texte sans codes I,U, S) ne sont pas imprimés. Le défaut, COMM = 1, provoque l'impression de ces commentaires.

WARN = Si WARN = 0, les messages d'avertissement de type "...W" ne sont pas imprimés (cfr. G. Les messages propres au programme CSCHECK). Le défaut, WARN = 1, permet l'impression de ces messages de type "...W".

SSOURCE = '*' ; cette instruction renvoie à une liste de (chemin et) noms de fichiers SOURCE après les options (à partir d'une autre ligne et sans apostrophes).
= 'chemin et nom de fichier' ; cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type SOURCE.

C. Exemples

1. Exemple de fichier d'exécution

```
call protan.bat csccheck bondsch.lis bondsch.sch 58
```

2. Exemples de fichiers de paramètres et options

Contenu du fichier "bondsch.sch":

```
PRINT = 0, COMM = 0, WARN = 1, SENT = '!.? ', SSOURCE = '*';
c:\protan\data.sou\bond\bond1.sou
c:\protan\data.sou\bond\bond2.sou
```

Le résultat sera une liste imprimée ne contenant que les "warnings" éventuels avec les messages correspondant. Les séparateurs de phrase considérés sont le point, le point d'exclamation, et le point d'interrogation.

D. Préparation des données

La seule préparation consiste à créer le ou les fichiers selon la syntaxe décrite au Chapitre IV, en respectant l'ordre logique des enregistrements 1 à 4 (*SOURCE-identification, commentaires libres, *ENDCOM, ces trois types d'enregistrement étant suivis du texte), l'ensemble pouvant éventuellement être scindé en plusieurs fichiers.

E. Autres informations

Tout traitement de données appelle une étape de vérification de ces données. Des données textuelles peuvent comporter, soit des erreurs dans l'orthographe des mots, soit des erreurs dans l'usage des caractères séparateurs de phrases, soit encore des erreurs dans les codes de segmentation. Si les premières erreurs ne sont pas détectées par le programme CSCHECK, elles n'en sont pas moins importantes. En effet, si le mot "soleil" a été orthographié "solel" par exemple, la comparaison "soleil/solel" (dans le programme CDWLOOK) entre le mot "soleil", présent éventuellement dans une des catégories d'un dictionnaire, et le mot "solel" du texte, donnera un résultat négatif.

En principe, le traitement d'un texte, quel que soit d'ailleurs le programme de traitement, doit se terminer par le message de "fin normale de travail":

(CSCH001I PROCESSING TERMINATED - NORMAL END OF JOB).

Dans le cas du programme CSCHECK, on n'aura ce type de message que si les seuls messages générés sont des messages d'information. Les messages d'avertissement donnent lieu au message final:

"CSCH002W PROCESSING TERMINATED - HOWEVER "WARNING" DIAGNOSTICS HAVE BEEN GENERATED".

F. Remarques

Néant

G. Messages propres au programme CSCHECK

*CSCH371E	illegal use of character or marker not closed
*CSCH372E	nonverbal marker not allowed in text
*CSCH373W	item will be truncated
*CSCH374E	column nn not empty
*CSCH375S	invalid int code
*CSCH376E	int sequence error
*CSCH377S	invalid unit code
*CSCH378E	unit sequence error
*CSCH379S	invalid speaker code
*CSCH380W	int sequence broken
*CSCH381W	unit sequence broken
*CSCH386S	invalid sentence separator (SENT)

XVIII. Le programme CSCUT

A. But du programme

CSCUT a pour but de créer un fichier-système de type "WORDS" nécessaire à tous les programmes ultérieurs. Le fichier-système "WORDS" transforme la triple segmentation du texte en interviews, unités, et locuteurs (I, U, et S) en une segmentation unique variant selon les valeurs des options BRKB, BRKM, SEGT, et CTRL. Ce fichier-système de type "WORDS" sera utilisé ultérieurement par tous les programmes qui analysent ou transforment le texte.

B. Exécution

1. Appel du programme

```
... CSCUT fn1 fn2 [fn3...]
```

a. où fn1 est le nom (filename) d'un fichier dont le type (filetype) est CSCUT en entrée; ce même fn1 sera le nom, en sortie, d'un fichier dont le filetype sera "WORDS", "PUNCHW", "SASWORD", "PUNCHS", ou "SASSTAT" (voir infra "Paramètres et options"); les fichiers de type "PUNCHW" ou "SASWORD" contiennent, sous forme perforée, la liste des mots triés, pouvant être traités par un logiciel d'analyse statistique, quelconque dans le cas de "PUNCHW", et SAS dans le cas de "SASWORD". Les fichiers de type "PUNCHS" et "SASSTAT" contiennent, sous forme perforée, les diverses statistiques de tri.

b. où fn2, fn3, etc. est (sont) le (les) nom(s) de fichier(s) de filetype "SOURCE", c'est-à-dire le ou les fichiers contenant le texte original dans son ordre séquentiel. Ces fichiers partiels peuvent être soumis condensés à la procédure CSCUT.

2. Entrée/sorties: En entrée, CSCUT nécessite, comme CSCHECK, le ou les fichiers partiels, c'est-à-dire le texte original. Mais en sortie, et par défaut, CSCUT crée un fichier-système de type "WORDS". Ce fichier "WORDS", lisible seulement par les programmes qui en ont besoin, contient notamment la liste alphabétique des mots différents contenus dans le corpus, avec leurs fréquences et d'autres statistiques de tri. Seules les options "TABLE=1" et "STAT=1" permettent une impression lisible de cette liste alphabétique et de ces statistiques de tri.

Optionnellement, CSCUT crée un fichier de filetype "PUNCHW" ou "SASWORD" contenant la liste des mots triés (ou encore un fichier de filetype "PUNCHS" ou "SASSTAT" contenant les statistiques de tri).

3. Paramètres et options:

BRKB = Cette option obligatoire détermine la segmentation finale du corpus. BRKB doit être suivi, entre apostrophes, de 1 à 3 codes d'interview, d'unité, ou de locuteur,

seuls ou en combinaison. On aura donc "BRKB='I'", ou "BRKB = 'U'", ou "BRKB = 'US'", ou l'une des autres combinaisons possibles des codes I, U, et S.

BRKM = Cette option définit le nombre de ruptures nécessaires pour créer un nouveau segment. Par défaut, *BRKM* est égal à 1 (et doit toujours être supérieur à zéro). Par exemple, si "BRKB = 'S'" et "BRKM = 2", on aura un nouveau segment chaque fois que deux locuteurs auront été enregistrés.

SEGT = Cette option, en combinaison avec la suivante, *CTRL*, régit la resegmentation du corpus (mais toujours dans les limites définies par *BRKB* et *BRKM*). On peut en effet avoir des raisons de ne pas utiliser la segmentation initiale du corpus en interviews, unités et locuteurs, et préférer par exemple une nouvelle segmentation par 200 mots, ou par 20 phrases, ou encore par 10 "marqueurs", si de tels marqueurs ont été insérés dans le texte.

SEGT peut prendre 3 valeurs:

- a. *SEGT* = 'W' (pour une resegmentation par mot);
- b. *SEGT* = 'S' (pour une resegmentation par phrase);
- c. *SEGT* = '#marqueur#' (pour une resegmentation par marqueur).

CTRL = Si *SEGT* est utilisé, *CTRL* doit être plus grand que zéro. C'est le facteur de multiplication de *SEGT*.

RESIDUE= Les valeurs que peut prendre l'option *RESIDUE* sont 'keep', 'drop', ou 'merge'. Dans le cas de 'keep' (défaut), le dernier segment est conservé tel quel lors de la segmentation par *MODULO* (voir plus loin) ou par <*SEGT* = 'w'>. Dans le cas de 'drop', ce dernier segment est simplement supprimé. Par contre, il est joint au segment précédent dans le cas de <*RESIDUE* = 'merge'>.

MODULO = Vaut 0 ou n, le défaut étant 0. Les valeurs minimales de *MODULO* sont 2 avec <*RESIDUE* = 'drop'> ou <*RESIDUE* = 'merge'>, et 3 avec <*RESIDUE* = 'keep'>. Cette option permet de démultiplier la segmentation initiale en un nombre n (valeur de *MODULO*) égal de nouveaux segments. Chacun de ces

nouveaux segments comporte un nombre strictement égal de mots à l'exception du dernier. Dans le cas de $\langle \text{RESIDUE} = \text{'drop'}$ ou $\text{'merge'} \rangle$, le diviseur du nombre total de mots (correspondant à chacun des anciens segments) est égal à la valeur affectée à MODULO ; dans le cas de $\langle \text{RESIDUE} = \text{'keep'} \rangle$, ce diviseur est égal à la valeur de MODULO moins 1, le dernier nouveau segment recevant la valeur du reste de la division. Par exemple: soit le nombre total de mots = 22 pour un (ancien) segment. Dans le cas de:

$\langle \text{MODULO} = 4, \text{RESIDUE} = \text{'keep'} \rangle$,

on aura:

	Nombre de mots par segment
nouveau segment 1	6
nouveau segment 2	6
nouveau segment 3	6
nouveau segment 4	4

Dans le cas de:

$\langle \text{MODULO} = 4, \text{RESIDUE} = \text{'drop'} \rangle$,

on aura:

	Nombre de mots par segment
nouveau segment 1	5
nouveau segment 2	5
nouveau segment 3	5
nouveau segment 4	5

Et dans le cas de:

$\langle \text{MODULO} = 4, \text{RESIDUE} = \text{'merge'} \rangle$,

on aura:

	Nombre de mots par segment
nouveau segment 1	5
nouveau segment 2	5
nouveau segment 3	5
nouveau segment 4	7

REGEN= Par défaut, $\langle \text{REGEN} = 0 \rangle$. Dans le cas où MODULO est plus grand que zéro, les nouveaux segments créés à l'intérieur de la segmentation initiale sont additionnés les uns aux autres $\langle \text{REGEN} = 1 \rangle$. Le nombre de segments générés est égal à la valeur de MODULO . Par exemple:

avant REGEN: (avec $\text{RESIDUE} = \text{'merge'}$)

nombre de mots/segment					
ancien segment 1,	nouveau segment 1,				
"	"	1	"	"	5
"	"	1	"	"	5
"	"	1	"	"	5
"	"	1	"	"	7

"	"	2	"	"	1	4
"	"	2	"	"	2	4
"	"	2	"	"	3	4
"	"	2	"	"	4	5

après **REGEN**, on aura:

			nombre de mots/segment
nouveau segment	1,		9
"	"	2	9
"	"	3	9
"	"	4	12

SENT = Comme pour **CSCHECK**, cette option définit les caractères séparateurs de phrases reconnus par le système. Les cinq caractères suivants < ; . ! ? : > sont les séparateurs de phrases *par défaut*; <SENT = ';.!?:'>, ou <SENT = '!'>, ou aucune mention de l'option **SENT** sont des formules équivalentes.

Note: Toujours entourer les caractères séparateurs de phrases d'une apostrophe de part et d'autre, sans y introduire de caractère blanc (~). En imposant les caractères à reconnaître par le système comme séparateurs de phrases, on se donne la possibilité de resegmenter souplement un texte (en combinaison avec les options **SEGT** = 'S' et **CTRL**). Ces caractères peuvent être choisis dans la liste des séparateurs de phrases fonctionnant déjà par défaut et dans la liste des séparateurs de mots, à l'exception du blanc (~), qui ne peut pas être utilisé, et de l'apostrophe (') qui, pour pouvoir être utilisée, doit être redoublée dans la définition de l'option. On aura par exemple <SENT = ';.!?:/'> (on ajoute la barre oblique (/) à la liste des séparateurs de phrases fonctionnant par défaut). On pourrait avoir aussi <SENT = '/'> (seule la barre oblique sera reconnue comme séparateur de phrases, tous les autres caractères étant ignorés, s'ils existent).

Enfin, on retiendra que le choix de ces séparateurs de phrases détermine la valeur de la longueur moyenne des phrases fournies avec l'option **"STAT"**.

Les valeurs de l'option **SENT** vont déterminer les résultats des indices de Gunning et de la longueur moyenne des phrases, ainsi que le tableau de la distribution des longueurs de phrases. Ces valeurs vont aussi déterminer les résultats d'une segmentation par phrase (avec **SEGT**). Enfin, elles vont déterminer les résultats du programmes **CDWJUXT**.

COMM = Si **COMM** = 0, les commentaires insérés par l'utilisateur dans le texte (lignes de texte sans codes I, U, S) ne sont pas imprimés. Le défaut, **COMM** = 1, provoque l'impression de ces commentaires.

TABLE= "TABLE = 1" imprime la table des mots triés en ordre alphabétique. "TABLE = 0", le défaut, inhibe l'impression de cette liste.

-
- STAT* = Le défaut de cette option, "STAT = 1", imprime les statistiques de tri, c'est-à-dire nombre de mots différents, nombre total de mots, et rapport type-occurrences --TTR--; cette option donne en outre une information sur la longueur moyenne (et écart-type) des phrases par segment (AV. SENT. LENGTH et SD SENT. LENGTH), la longueur moyenne (et écart-type) des mots par segment (AV. WORD LENGTH et SD WORD LENGTH), et le pourcentage de mots de 9 caractères et plus (par segment) (% WORDS L. >= 9). L'indice de lisibilité de Gunning¹⁶ est obtenu par la formule:
- $$[(\text{AV. SENT. LENGTH} + \% \text{ WORDS L.} \geq 9) * 0.4].$$
- Un indice élevé est le signe d'un texte difficile à lire (de 6 à 8, texte lisible, de 12 à 15, texte difficile à lire). "STAT = 0" inhibe l'impression de ces informations.
- Une dernière information (inconditionnelle, c'est-à-dire non liée à "STAT = 1 ou 0") concerne la distribution des longueurs de phrases par groupes de 5 mots, depuis les phrases de 1 à 5 mots, de 6 à 10, etc., jusqu'aux phrases de 121 mots et plus (Reimer, 1992).
- PSTAT* = "PSTAT = 1" perfore en plus le contenu des statistiques de tri imprimées par le paramètre "STAT = 1", en créant un fichier de type "PUNCHS". Le défaut, "PSTAT = 0", inhibe la création de ce fichier. Pour le format de perforation de ce fichier de type "PUNCHS", voir "Remarques" infra.
- PRINT* = Par défaut, ("PRINT = 0"), le texte n'est pas imprimé. "PRINT = 1" permet l'impression du fichier-source. Si en outre l'option "NARR = 1" (voir plus loin) a été fournie, PRINT = 1 déclenche l'impression du fichier-source avec indication du numéro de séquence de chaque item, y compris tous les types de séparateurs, marqueurs, etc. L'option PRINT = 2 déclenche la même impression, mais uniquement avec comptage des mots (à l'exclusion des séparateurs, marqueurs, etc.).
- PUNCH* = "PUNCH = 1" perfore en plus le contenu de la liste imprimée par le paramètre "TABLE = 1", en créant un fichier de type "PUNCHW". Le défaut, "PUNCH = 0", inhibe la création de ce fichier. Pour le format de perforation de ce fichier de type "PUNCHW", voir "Remarques" infra.

¹⁶ Cet indice de lisibilité de Gunning (1952) est repris de Daoust (1992, p. 127). Sur la lisibilité, voir aussi Richeaudeau (1969) et Mailloux, Johnson, Fisher, et Pettibone (1995).

WORD = Par défaut (WORD = 1), le programme CSCUT crée toujours un fichier-système de type "WORDS" qui sera utilisé par tous les programmes qui analysent ou transforment le texte. "WORD = 0" inhibe la création de ce fichier.

NARR = Par défaut ("NARR = 0"), le fichier de type "WORDS" ne permet pas de retrouver l'ordre séquentiel initial des mots dans le texte. L'option "NARR = 1" ajoute au fichier de type "WORDS" des attributs qui permettent ultérieurement, lors de l'exécution des programmes CDWLOOK, CFWKWIC, CWEDIT, CWKWIC, CWPAT, et CWSELECT, de reconstituer l'ordre séquentiel et la valeur initiale des mots du texte; ces attributs sont utilisés, dans les programmes CDWLOOK et CWSELECT, par les options "PRINTNA = 1", "PUNCH = 3", et "PUNCH = 4" (voir ces options dans les programmes CDWLOOK et CWSELECT).

C. Exemples

1. Exemples de procédures d'appel

a. " ... CSCUT PSYCHO NEURO".

La procédure CSCUT fait d'abord appel au fichier "PSYCHO CSCUT" contenant les paramètres et options, et ensuite au fichier "NEURO SOURCE" qui est le texte soumis à l'analyse.

b. " ... CSCUT PSYCHO1 NEURO1 NEURO2 NEURO3".

Même chose que ci-dessus, le texte étant en outre scindé en 3 fichiers partiels appelés "NEURO1 SOURCE", "NEURO2 SOURCE", et "NEURO3 SOURCE".

c. Voir aussi le point "Remarques" du programme CSCHECK concernant l'usage d'une procédure de type "EXEC" pour l'appel des procédures de PROTAN.

2. Exemples de fichiers de paramètres et options

a. Contenu du fichier "PSYCHO CSCUT" (entre < et >):

```
<TABLE = 1, STAT = 1, PRINT = 0, PUNCH = 0, WORD = 1, BRKM = 1, BRKB = 'U';>
```

On demande la liste des mots en ordre alphabétique avec les statistiques de tri, sans impression des textes originaux, ni perforation des mots triés; on demande la création d'un fichier de type "WORDS", avec une segmentation sur les unités selon un facteur de combinaison de 1. En réalité, on a fait usage de la plupart des options par défaut, et la commande aurait pu s'écrire plus simplement <BRKB = 'U';>, qui est la seule option obligatoire.

b. Contenu du fichier "PSYCHO1 CSCUT" (entre < et >):

```
<CTRL = 3, SEGT = '#10M#', BRKB = 'I';>
```

Toutes les autres options étant prises par défaut, on resegmente au sein de chaque interview par groupes de 3 marqueurs "#10M#" --qui pourrait vouloir dire "10 minutes écoulées dans le texte depuis le dernier marqueur"--: On aura donc créé un nouveau segment toutes les 30 minutes.

c. Contenu du fichier "PSYCHO2 CSCUT" (entre < et >):

```
<BRKB = 'I', SEGT = 'S', CTRL = 1, SENT = '/';>
```

Chaque interview sera segmentée par phrase; les phrases sont délimitées par les barres obliques.

D. Préparation des données

Une segmentation ou resegmentation intelligente va dépendre de la manière dont le corpus aura été divisé en interviews, unités, et locuteurs ou, en ce qui concerne la resegmentation, des significations et répartition des marqueurs dans le texte, ou des caractères imposés pour la séparation des phrases. Le contenu ou le sens des interviews, unités, locuteurs, ou marqueurs sera ce que l'analyste décidera. Un marqueur "#T#" pourra par exemple être placé dans un protocole à chaque intervention du thérapeute. Dans ce même protocole, l'analyste pourrait aussi se dire que le mode interrogatif est particulièrement significatif d'une rupture dans un flux d'idées, ou d'un processus psychologique déterminé, et décider en conséquence d'insérer un marqueur "#Q#" chaque fois que le patient utilise le mode interrogatif dans son discours.

E. Autres informations

Aucune

F. Remarques

Le format de perforation du fichier de type "PUNCHW" que crée CSCUT est caractérisé par un LRECL de 40. Un fichier de type "PUNCHW" est utilisable par n'importe quel logiciel d'analyse statistique.

Les lignes qui suivent décrivent les fichiers de type "PUNCHW" d'abord et "PUNCHS" ensuite:

```
PUNCHW
créé par CSCUT, CRWSTRIP, CWKWOC, et CWEDIT
(LRECL = 40)
```

1er enregistrement:

col	1 à 8	mot-clé "*PUNCHW~"
	9 à 16	identification des fichiers de sortie
	17 à 24	date de création des fichiers de sortie
	25 à 32	heure de ...(idem)
	33 à 40	vierges

2me enregistrement:

col	1 à 8	vierges
	9 à 16	identification du fichier SOURCE/WORDS en entrée
	17 à 24	date de création du fichier WORDS en entrée, vierges si c'est un fichier SOURCE
	25 à 32	heure de ...(idem)

33 à 40 vierges

3me enregistrement:

col 1 à 8 vierges

9 à 16 identification du fichier ROOTS ou CWEDIT
 utilisé (pour CRWSTRIP ou CWEDIT), vierges
 CSCUT, "CWKWOC" pour CWKWOC

17 à 24 date de création du fichier ROOTS ou CWEDIT
 pour CRWSTRIP ou CWEDIT, sinon vierges

25 à 32 heure de ... (idem)

33 à 40 vierges

4me enregistrement:

col 1 à 5 mot-clé "WORD~"

6 à 10 mot-clé "~POPT"

11 à 15 nombre d'options de traitement fournies à CSCUT

16 à 20 mot-clé "~GCOM"

21 à 25 nombre de lignes de commentaires généraux
 (le nombre de lignes de 80 caractères fournies
 par l'utilisateur est exprimé ici en un nombre
 double de lignes de 40 caractères)

26 à 30 mot-clé "~~OPT"

31 à 35 nombre d'options de traitement fournies au
 programme ayant généré ce fichier (toujours
 zéro pour CSCUT)

36 à 40 vierges

5me enregistrement:

col 1 à 5 vierges

6 à 10 mot-clé "~NSEG"

11 à 15 nombre de segments générés

16 à 40 vierges

6me enregistrement:

col 1 à 40 les 40 premiers caractères du commentaire
 principal du fichier SOURCE (celui de
 l'enregistrement *SOURCE)

7me enregistrement:

col 1 à 24 les 24 derniers caractères du commentaire
 principal

25 à 40 vierges

les POPT enregistrements suivants:

col 1 à 40 une option sous la forme XXX=YYY;

les GCOM enregistrements suivants:

col 1 à 40 les 40 caractères d'un demi commentaire
 général

les OPT enregistrements suivants:

col 1 à 40 une option sous la forme XXX=YYY;

enregistrements suivants (triés par ordre alphabétique des mots, et
des segments à l'intérieur de chaque mot): chaque variable occupe 7
positions (sauf le mot -WORD- qui en occupe 20) et est séparée de la
suivante par un caractère de tabulation. Soit, pour chaque
enregistrement:

- mot (variable WORD)
- numéro du segment (variable SEG)
- fréquence du mot dans le segment (variable SF)
- fréquence totale du mot dans l'ensemble des
segments (variable TF),

n'est présente qu'en regard du dernier segment d'occurrence du mot; pour les segments précédents, la zone contient zéro.

Et passage à l'enregistrement suivant pour le 2ième mot, etc.

Le format de perforation du fichier de type "PUNCHS" créé par "PSTAT = 1" se présente, lui, comme suit:

PUNCHS
créé par PSTAT dans le programme CSCUT
(LRECL = 80)

1er enregistrement:

col 1 à 8	mot-clé "*PUNCHS~"
9 à 16	identification des fichiers de sortie
17 à 24	date de création des fichiers de sortie
25 à 32	heure de ...(idem)
33 à 80	vierges

2me enregistrement:

col 1 à 5	mot-clé "STAT~"
6 à 10	mot-clé "~POPT"
11 à 15	nombre d'options de traitement fournies à CSCUT
16 à 20	mot-clé "~GCOM"
21 à 25	nombre de lignes de commentaires généraux
26 à 30	mot-clé "~~OPT"
31 à 35	nombre d'options de traitement fournies au programme ayant généré ce fichier (toujours zéro pour CSCUT)
36 à 40	vierges
41 à 45	mot-clé "~NSEG"
46 à 80	nombre de segments générés

3me enregistrement:

col 1 à 80	le commentaire principal du fichier SOURCE (celui de l'enregistrement *SOURCE)
------------	--

les POPT enregistrements suivants:

col 1 à 80	une option sous la forme XXX=YYY;
------------	-----------------------------------

les GCOM enregistrements suivants:

col 1 à 80	un commentaire général
------------	------------------------

les OPT enregistrements suivants:

col 1 à 80	une option sous la forme XXX=YYY;
------------	-----------------------------------

enregistrements suivants (triés par ordre du numéro de segment):
chaque variable occupe 9 positions et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

	-	numéro du segment (variable SEG)
	-	nombre de mots différents (variable WD)
	-	nombre total de mots (variable WT)
	-	rapport types-occurrences (variable TTR) exprimé par une proportion à 3 décimales
décimales)	-	longueur moyenne des phrases (variable SENTL) (à 3 décimales)
décimales)	-	longueur moyenne des mots (variable WORDL) (à 3 décimales)
PERC9) (à 3 décimales)	-	% des mots de 9 caractères ou plus (variable PERC9) (à 3 décimales)
3 décimales).	-	indice de lisibilité de GUNNING (variable GUNN) (à 3 décimales).

Et passage à l'enregistrement suivant pour le 2ième segment, etc.

Notons que ce fichier "PUNCHS" (avec les diverses statistiques qu'il contient) n'est disponible qu'au stade du programme CSCUT.

G. Messages propres au programme CSCUT

*CSCU401S	RESIDUE type invalid
*CSCU402S	invalid segmentation counter limit (CTRL)
*CSCU403S	invalid break counter limit (BRKM)
*CSCU404S	invalid break code (BRKB)
*CSCU405S	duplicate break code (BRKB)
*CSCU406S	no break code specified (BRKB)
*CSCU407S	seg will be > 99999
*CSCU408S	syntax error in source record
*CSCU409S	MODULO value too low
*CSCU410S	invalid sentence separator (SENT)

A noter que l'erreur *CSCU408S correspond au type d'erreur E(rror) ou S(evere error) dans le programme CSCHECK. Cette erreur entraîne en même temps l'arrêt du programme.

XIX. Le programme CSEDIT

A. But du programme

Un fichier-source doit être édité lorsqu'il y a des modifications à y apporter. Dans le système PROTAN, on ne peut apporter des modifications au texte que par une édition en CMS (si on excepte les modifications qu'opèrent les dictionnaires de formes nominales, voir Chapitre XXXII. Annexes). Par contre, lorsqu'on doit apporter des modifications systématiques au contenu des colonnes 73 à 80, le programme CSEDIT est sans doute plus efficace qu'une édition, vite fastidieuse, en CMS ou à l'aide de tout autre éditeur.

B. Exécution

1. Appel du programme

```
... CSEDIT fn1 fn2 [fn3...]
```

a. où fn1 est le nom, en entrée, d'un fichier, contenant les paramètres et les options (cfr. infra), dont le filetype est CSEDIT; en sortie, ce même nom fn1 sera le nom d'un fichier de filetype "SOURCE" que crée le programme CSEDIT.

b. où fn2, fn3, etc... sont les noms des fichiers partiels à éditer.

2. Entrées/sorties: En entrée, CSEDIT demande d'abord un fichier fn1 CSEDIT: Ce fichier fn1 contient en premier lieu les paramètres et options du programme; fn1 contient en deuxième lieu les symboles syntaxiques des fichiers-source¹⁷ qui seront récupérés par le programme et placés en tête du nouveau fichier-source; en troisième lieu, dans fn1 doit figurer le contenu des champs (100 enregistrements au maximum) sur lesquels vont porter l'édition (interview, unité, et locuteur).

En entrée toujours, CSEDIT demande ensuite le ou les fichiers partiels fn2, fn3, etc... qui vont faire l'objet de l'édition.

En sortie, CSEDIT crée un seul fichier-source édité à partir du ou des fichiers-source partiels reçus en entrée; ce nouveau fichier-source aura pour nom le nom fn1 du fichier "fn1 CSEDIT" qui contient les paramètres.

3. Paramètres et options:

COMM = Si COMM = 0, les commentaires insérés par
 l'utilisateur dans le texte (lignes de

¹⁷ Au minimum, "*SOURCE-" avec l'identification du fichier-source et, à la ligne suivante, "*ENDCOM", signalant la fin des commentaires, même si aucun commentaire n'est inséré entre le premier et le second symbole.

texte sans codes I,U, S) ne sont pas imprimés. Le défaut, COMM = 1, provoque l'impression de ces commentaires.

PRINT = A la demande ("PRINT = 1"), on peut faire imprimer le fichier édité résultant du programme CSEDIT. Par défaut, ("PRINT = 0"), ce fichier n'est pas imprimé.

OUT = Puisque le but de CSEDIT est d'apporter des modifications à un fichier, il y a création d'un nouveau fichier-source par défaut ("OUT = 1"). "OUT = 0" inhibe cette création.

EDIT = Cette option, obligatoire de même que la suivante, "FIELD =", gère le type d'édition demandé.

"EDIT = 'PASS'" permet de sélectionner des champs, isolément ou en combinaison avec d'autres champs (voir option "FIELD =").

"EDIT = 'DROP'" permet, au contraire du précédent, d'ignorer des champs lors de la création du nouveau fichier-source. Ici aussi, on peut ignorer des champs homogènes (I, U, ou S) ou des combinaisons de champs (IU, IS, IUS, US).

"EDIT = 'RENUM'", enfin, permet de renuméroter certains champs ou combinaisons de champs.

FIELD = Cette option obligatoire précise le ou les champs de l'édition, soit:

```
"FIELD= 'I' "  
        = 'U' "  
        = 'S' "  
        = 'IU' "  
        = 'IS' "  
        = 'US' "  
        = 'IUS'
```

Cette série d'options est suivie 1) des symboles syntaxiques du fichier-source créé en sortie et 2) des champs d'édition (max 100).

C. Exemples

1. Exemple de procédure d'appel

a. " ... CSEDIT DOUZEDT TAT1 TAT2 TAT3".

La procédure fait appel au fichier "DOUZEDT CSEDIT" qui contient les paramètres et options, les symboles syntaxiques du fichier-source à créer, et le contenu des champs d'édition; les noms TAT1, TAT2, et TAT3 sont les noms des fichiers partiels à éditer, le premier fichier TAT1 contenant en outre les symboles syntaxiques identifiant le fichier-source non édité, l'ancienne version donc si on veut.

2. Exemples de fichiers de paramètres et options

a. Contenu du fichier "DOUZEDT CSEDIT" (entre < et >):

```
<
EDIT= 'PASS', FIELD= 'I';
*SOURCE-DOUZEDT          COM: FICHIER EDITE
COMMENTAIRES: SELECTION DES INTERVIEWS 1, 3, 5, et 7 SUR LES 8
INTERVIEWS PRESENTS.
IDENTIFICATION ANTERIEURE DU TEXTE : "DOUZE"
*ENDCOM
001
003
005
007
>
```

D'un corpus (ancien fichier "DOUZE SOURCE") divisé en 8 interviews, on ne retient que les 4 interviews n° 1, 3, 5, et 7. Les interviews retenus doivent être placés en colonnes 1 à 3 très précisément. En fait, les colonnes 1 à 8 du fichier "fn1 CSEDIT" reproduisent les colonnes 73 à 80 du fichier "fn2 (fn3, ...) SOURCE". Le nouveau fichier-source est signalé par un nouvel identificateur, "DOUZEDT", qui permet de créer le fichier sous un nom différent de l'ancien "DOUZE SOURCE".

b. Contenu d'un fichier "DOUZEDT1 CSEDIT" (entre < et >):

```
<
EDIT = 'DROP', FIELD = 'IS';
*SOURCE-DOUZEDT1          ANCIEN FICHIER "DOUZEDT SOURCE"
COMM: ON DESIRE ELIMINER DU FICHIER TOUS LES TEXTES QUI
CONCERNENT LE LOCUTEUR B DANS LES INTERVIEWS 2, 4, 6, ET 8.
*ENDCOM
002      B
004      B
006      B
008      B
>
```

"DOUZEDT1" est le nouvel identificateur du fichier d'où sont exclus les interviews 2, 4, 6, et 8 où le locuteur B est présent. L'indication d'interview figure en colonnes 1 à 3, celle de locuteur, en colonne 8.

c. Contenu d'un fichier "DOUZEDT2 CSEDIT" (entre < et >):

```

<
EDIT = 'RENUM', FIELD = 'I';
*SOURCE-DOUZEDT2          ANCIEN FICHIER DOUZEDT1
COMM:ON VEUT RENUMEROTER LES INTERVIEWS 2, 4, 6, ET 8.
*ENDCOM
002      001
004      002
006      003
008      004
>

```

Les interviews 2, 4, 6, et 8 (en colonnes 1 à 3) seront renumérotés 1, 2, 3, et 4 (en colonnes 11 à 13) respectivement, tout le reste demeurant inchangé.

D. Préparation des données

Comme on l'a signalé dans les exemples ci-dessus, les colonnes 73 à 80 d'un fichier-source sont, dans ce programme d'édition, représentées:

1. par les colonnes 1 à 8 pour les valeurs "PASS" et "DROP" de l'option "EDIT =",

2. par les colonnes 1 à 8 et 11 à 18 pour la valeur "RENUM" de l'option "EDIT =".

On aura donc:

```

colonnes.....
0 0 0 0 0 0 0 0
1 2 3 4 5 6 7 8

x x x          ...pour PASS/DROP de I
          x x x  ...pour PASS/DROP de U
                  x  ...pour PASS/DROP de S
x x x          x  ...pour PASS/DROP de I et S
x x x  x x x    ...pour PASS/DROP de I et U
          x x x x ...pour PASS/DROP de U et S
x x x  x x x x  ...pour PASS/DROP de I, U, et S

```

```

colonnes
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8

x x x          x x x          x x x          ...pour RENUM de I
          x x x          x x x          ...pour RENUM de U
                  x          ...pour RENUM de S
x x x  x x x    x x x  x x x    ...pour RENUM de I et U
x x x          x x x          x          ...pour RENUM de I et S
          x x x x          x x x x    ...pour RENUM de U et S
x x x  x x x x  x x x  x x x x    ...pour RENUM de I, U,
                                   et S

```

E. Autres informations

Aucune.

F. Remarques

A noter que le programme CSSORT conserve toutes les fonctionnalités du programme CSJOIN. Il conviendrait donc, par exemple, de paier les **source-* et les **endcom* si on venait à en insérer en tête de chaque nouveau texte.

G. Messages propres au programme CSEDIT

*CSED421S	edit type invalid or not specified
*CSED422S	invalid field code
*CSED423S	duplicate field code
*CSED424S	no field code specified
*CSED425E	more then 100 field records
*CSED426S	syntax error in field record
*CSED427S	syntax error in source record
*CSED428S	no field record specified
*CSED429U	detection of a <i>*source</i> after a <i>*source</i> to <i>*endcom</i> sequence

XX. Le programme CSJOIN

A. But du programme

CSJOIN va concaténer en un seul fichier des fichiers de noms différents, mais de types ("SOURCE") identiques.

B. Exécution

1. Appel du programme

```
... CSJOIN fn1 fn2 [fn3 ...]
```

a. où fn1 est le nom, en entrée, d'un fichier (qui contient les paramètres et les options, cfr. infra) dont le filetype est CSJOIN; en sortie, ce même nom fn1 sera le nom unique du fichier de filetype "SOURCE" sous lequel seront concaténés les fichiers de noms différents.

b. où fn2, fn3, etc... sont les noms des fichiers à concaténer sous le nom fn1.

2. Entrées/sorties: En entrée, CSJOIN appelle d'une part un fichier "fn1 CSJOIN" contenant d'abord les paramètres et options, et ensuite les symboles syntaxiques du nouveau fichier--source. (Au minimum, "*SOURCE-" avec l'identification du fichier, et, à la ligne suivante, "*ENDCOM~", qui signale la fin des commentaires, même si on n'a pas inséré de commentaires entre "*SOURCE" et "*ENDCOM~"). Ces symboles syntaxiques sont récupérés par le programme et placés en tête des fichiers concaténés.

Toujours en entrée, CSJOIN appelle d'autre part les différents fichiers-source à concaténer sous le nom de "fn1 SOURCE".

En sortie à présent, CSJOIN crée un fichier-source unique à partir des différents fichiers dont il a reçu les noms en entrée. Ce fichier-source unique a pour nom le nom fn1 du fichier "fn1 SOURCE" contenant paramètres et options.

3. Paramètres et options:

COMM = Si *COMM* = 0, les commentaires insérés par l'utilisateur dans le texte (lignes de texte sans codes I,U, S) ne sont pas imprimés. Le défaut, *COMM* = 1, provoque l'impression de ces commentaires.

PRINT = Par défaut ("*PRINT* = 0"), les divers fichiers-source ne sont pas imprimés une nouvelle fois. "*PRINT* = 1" permet néanmoins d'obtenir l'impression de tous les fichiers à concaténer.

OUT = Puisque le but de CSJOIN est de concaténer des fichiers, il y a création d'un nouveau fichier-source par défaut ("*OUT* = 1"). "*OUT* = 0" ne produit pas de nouveau fichier-source.

FILL = Si *FILL*=1, les champs réservés aux zones d'interview (I), d'unité (U), et de speaker (S) sont remplacées par des valeurs

arbitraires définies par les options **fillerI**, **fillerU**, et **fillerS**. Si un fichier-source est créé sans les codes d'interview, d'unité et de speaker, l'option FILL superpose au fichier-source le champ prévu pour I, U, et S.
Par défaut, FILL=0, et les champs compris entre les zones 73 à 80 ne sont pas altérés.

FILLERI = par défaut, la zone 73 à 75 est remplie avec les codes '***',

FILLERU = par défaut, la zone 77 à 79 est remplie avec les codes '%%%',

FILLERS = par défaut, la zone 80 est remplie avec le caractère '@'.

Chacun des codes ***, %%, ou @ peut ensuite être modifié dans le fichier-source par un éditeur de texte.

Les symboles syntaxiques du fichier-source à constituer en sortie suivent cette série d'options (voir Entrées/sorties).

C. Exemples

1. Exemple de procédure d'appel

```
" ... CSJOIN DOUZE TAT1 TAT2 TAT3"
```

TAT1, TAT2, et TAT3 sont les différents fichiers de type "SOURCE" à concaténer sous un seul nom qui sera "DOUZE". (Disons qu'il s'agit des réponses de patients à la planche n° 12 du T.A.T.). Le fichier "DOUZE CSJOIN" contient alors les paramètres et options de la procédure.

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "DOUZE CSJOIN" (entre < et >):

```
<
*SOURCE-DOUZE          TRAVAIL DE STAGE DE MONSIEUR X
REPONSES DE 15 PATIENTS A LA PLANCHE NO. 12 DU  T.A.T...PASSATION
PAR MONSIEUR X LE ...: GROUPE EXPERIMENTAL
*ENDCOM
>
```

Les options de CSJOIN sont prises par défaut, il en résultera un fichier "DOUZE SOURCE" contenant 15 protocoles de T.A.T. Les symboles syntaxiques qui identifient un fichier-source (*SOURCE-avec identification, commentaires éventuels, et "*ENDCOM~") se trouveront en tête des fichiers concaténés.

D. Préparation des données

Lorsque les fichiers à concaténer sont de type "SOURCE" et ont été écrits selon la syntaxe propre aux fichiers de ce type, aucune préparation n'est nécessaire pour l'exécution de la procédure CSJOIN. Les fichiers ainsi concaténés en un seul fichier

peuvent toujours être soumis au programme CSCHECK pour vérification.

E. Autres informations

A noter que le programme CSSORT conserve toutes les fonctionnalités du programme CSJOIN. Il conviendrait donc, par exemple, de paier les **source-* et les **endcom* si on venait à en insérer en tête de chaque nouveau texte.

F. Remarques

CSJOIN met seulement ensemble les fichiers qu'on lui propose, sans y apporter de modifications; ces dernières devraient être opérées soit directement par un travail d'édition en CMS, soit par les programmes CSEDIT et/ou CSSORT.

G. Messages propres au programme CSJOIN

<pre>*CSJO441U detection of a *source after a *source to *endcom sequence</pre>
--

XXI. Le programme CSSORT

A. But du programme

Pour une exécution correcte du programme CSCHECK, il faut que les textes soient rangés en ordre croissant d'interviews, et d'unités à l'intérieur des interviews. Cet ordre croissant ne doit pas nécessairement être continu. Il peut cependant arriver que, soit par une erreur lors de l'introduction des données, soit par l'incidence d'une procédure CSJOIN ou CSEDIT, les textes ne soient plus en ordre croissant d'interviews et d'unités. A ce moment, la procédure CSSORT permet de remettre les textes dans l'ordre voulu. Comme les autres programmes de type CS- (CSCHECK, CSJOIN et CSEDIT), CSSORT va donc appeler en entrée un ou des fichiers de type "SOURCE", et créer, en sortie, un seul fichier de type "SOURCE" lui aussi.

B. Exécution

1. Appel du programme

```
... CSSORT fn1 fn2 [fn3 ...]
```

a. où fn1 est le nom (en entrée) d'un fichier contenant les paramètres et les options (cfr. infra), fichier dont le type est CSSORT; en sortie, ce même nom fn1 sera le nom du nouveau fichier de filetype "SOURCE" créé par CSSORT.

b. où fn2, fn3, etc... sont les noms des fichiers partiels à éditer.

2. Entrées/sorties: En entrée, CSSORT demande d'abord un fichier fn1 CSSORT; ce fichier fn1 contient en premier lieu les paramètres et options du programme; fn1 contient en second lieu les symboles syntaxiques¹⁸ du nouveau fichier-source; ces symboles syntaxiques seront récupérés par le programme et placés en tête du nouveau fichier-source de nom fn1.

En entrée encore, CSSORT demande ensuite un ou des fichiers partiels fn2, fn3, etc... dont les colonnes 73 à 80 vont faire l'objet du tri.

En sortie, CSSORT crée un seul fichier-source de filename fn1 à partir du ou des fichiers-source partiels fn2, fn3, ... reçus en entrée.

3. Paramètres et options:

COMM = Si *COMM* = 0, les commentaires insérés par l'utilisateur dans le texte (lignes de texte sans codes I,U, S) ne sont pas im-

¹⁸ Au minimum, "*SOURCE-" avec l'identification du nouveau fichier-source, et, à la ligne suivante, "*ENDCOM~", signalant la fin des commentaires, même si aucun commentaire n'est présent entre "*SOURCE-" et "*ENDCOM~".

- primés. Le défaut, `COMM = 1`, provoque l'impression de ces commentaires.
- `LIST1=` A la demande ("`LIST1 = 1`"), CSSORT imprime le fichier-source dans son état avant le tri. Par défaut ("`LIST1 = 0`"), il n'y a pas impression, sauf des commentaires et messages.
- `LIST2=` A la demande ("`LIST2 = 1`"), CSSORT imprime le fichier-source tel qu'il apparaît après le tri. Par défaut ("`LIST2 = 0`"), il n'y a pas impression.
- `OUT=` Par défaut ("`OUT = 1`"), il y a toujours constitution d'un fichier-source. "`OUT = 0`" inhibe la création de ce fichier-source.
- `SORTT=` Par défaut ("`SORTT = 'IU'`"), les textes seront triés par interviews, et par unités à l'intérieur des interviews. Mais d'autres types de tri sont possibles, isolés ou en combinaison. On pourra donc avoir les combinaisons suivantes:

```
SORTT = 'I'
        'U'
        'S'
        'IU'
        'IS'
        'UI'
        'US'
        ...
        'IUS'
        'SUI'
```

Note: Les symboles syntaxiques du nouveau fichier-source se placent après cette série d'options.

C. Exemples

1. Exemple de procédure d'appel

a. " ... CSSORT DOUZE TAT1 TAT2"

La procédure fait appel au fichier "DOUZE CSSORT" qui contient paramètres et options, ainsi que les symboles syntaxiques du nouveau fichier-source "DOUZE SOURCE" qui sera composé des fichiers "TAT1 SOURCE" et "TAT2 SOURCE".

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "DOUZE CSSORT" (entre < et >):

```
<;
*SOURCE-DOUZE
*ENDCOM
>
```

Supposons que le fichier "TAT1 SOURCE" soit composé de:

```

7.....8
colonnes...34567890
texte no. 1... 001 001A
texte no. 2... 002 001B
```

et le fichier "TAT2 SOURCE", de:

```

7.....8
colonnes...34567890
texte no. 3... 001 002A
texte no. 4... 002 002B
```

L'option "SORTT = 'IU'" étant prise par défaut, les deux fichiers TAT1 et TAT2 seront réunis en un seul fichier "DOUZE SOURCE" où on trouvera, dans l'ordre:

```

7.....8
colonnes 34567890
texte no. 1... 001 001A
texte no. 3... 001 002A
texte no. 2... 002 001B
texte no. 4... 002 002B
```

D. Préparation des données

Aucune.

E. Autres informations

Aucune.

F. Remarques

A noter que le programme CSSORT conserve toutes les fonctionnalités du programme CSJOIN. Il conviendrait donc, par exemple, de paier les **source-* et les **endcom* si on venait à en insérer en tête de chaque nouveau texte.

G. Messages propres au programme CSSORT

*CSS0461S	invalid sort type
*CSS0462S	duplicate sort type
*CSS0463S	syntax error in source record
*CSS0464S	no sort type specified
*CSS0465U	detection of a <i>*source</i> after a <i>*source</i> to <i>*endcom</i> sequence

XXII. Le programme CWADD

A. But du programme

Il est parfois utile d'ajouter au texte des indicateurs non-verbaux qui ne figurent pas dans la version originale. Par exemple, il serait pertinent d'ajouter à des discours politiques distribués sur plusieurs années des indicateurs économiques ou sociaux comme le taux de chômage, le PNB, ou le nombre de journées de grève par an. Il n'est cependant pas possible de pratiquer sur un fichier-système "WORDS" l'opération chirurgicale qui consisterait à y greffer des informations supplémentaires. Le programme CWADD a été conçu pour pouvoir greffer sur un fichier-système "WORDS" toute information utile: cette greffe se fait selon des règles précises en raison de la complexité d'un fichier-système.

B. Exécution

1. Appel du programme

```
... CWADD fn1 fn2
```

a. où fn1 est le nom d'un fichier de type CWADD qui contient les paramètres et options de la (ou des) variables qu'on veut ajouter au fichier "WORDS".

b. où fn2 est le filename du fichier de type "WORDS" créé par CSCUT, CRWSTRIP, CWEDIT ou CWADD. Ce fichier est le corpus à traiter, auquel on greffe maintenant une ou plusieurs variables supplémentaires avec leur fréquence par segment.

2. Entrées/sorties: Deux entrées sont nécessaires pour le programme CWADD, soit le fichier de paramètres et options ainsi que le nom de l'ancien fichier-système "WORDS" (le fichier à greffer) qui sera modifié à l'issue du programme CWADD.

A part une liste imprimée de longueur variable, l'autre sortie est le nouveau fichier-système "WORDS" qui est à présent enrichi d'une ou plusieurs variables supplémentaires.

3. Paramètres et options:

STAT = "STAT = 1" récapitule les définitions de segment demandées par l'utilisateur et donne les statistiques de tri, c'est-à-dire, par segment demandé, le nombre de mots différents et le nombre total de mots. "STAT = 0" (le défaut) inhibe cette impression.

PRINT = Par défaut, on n'imprime pas les paramètres ("PRINT = 0", défaut). "PRINT = 1" déclenche l'impression des paramètres de CWADD.

WORD = On s'attend toujours à ce qu'un nouveau fichier-système "WORDS" sorte de CWADD et donc, par défaut, "WORD = 1". Si "WORD = 0", le fichier "WORDS" n'est pas constitué.

Les options ci-dessus se terminent normalement par un ";" et sont suivies des paramètres qui vont formater chaque nouvelle variable dans le fichier-système "WORDS" qui est en train d'être constitué. Ces paramètres se composent de 4 parties:

- (1) en colonnes 1 à 20: l'**identification** de la variable. Cette identification commence par les symboles "##" suivis d'un nom approprié, par exemple, "##sept". Lorsque cette zone est vide, le programme prend la valeur présente à la ligne précédente;
- les colonnes 21 à 23 restent vides;
- (2) les colonnes 24 à 28 contiennent les **numéros des segments** où la nouvelle variable est présente (avec une certaine fréquence);
- les colonnes 29 à 31 restent vides;
- (3) les colonnes 32 à 39 sont réservées à la **fréquence** de la variable (par segment);
- les colonnes 40 à 42 restent vides à nouveau;
- (4) et enfin, la zone 43 à 80 est réservée à des **commentaires** éventuels.

On répète cette définition de paramètres autant de fois qu'il y a de variables à ajouter.

C. Exemples

1. Exemple de procédure d'appel

a. "... CWADD threat JAP".

Le fichier "threat CWADD" contient les options et les paramètres qui vont permettre d'ajouter au fichier "JAP WORDS" une ou plusieurs variables.

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "threat CWADD" (entre < et >, les indications en "subscript" ne font pas partie des paramètres):

```
< print = 1, stat = 1, word = 1,
  swords = 'c:\usr\jap\japrhs.wds',
  swordso = 'c:\usr\jap\japwad.wds';
```

1 à 20: ident.	24 à 28: segments	32 à 39: fréquences	43 à 80 : commentaire
##SEPT	004	34	1920
	005	29	1921
	006	26	1922
	007	25	1923
	008	23	1924
	009	22	1925
	010	22	1926
	011	23	1927
	012	26	1928
	013	49	1929

014	56	1930
015	59	1931
016	60	1932
017	55	1933
018	50	1934
019	46	1935
020	45	1936
021	47	1937
022	47	1938
023	51	1939
024	54	1940
025	61	1941
026	61	1942
027	56	1943
028	52	1944
029	45	1945
030	35	1946
031	36	1947
032	37	1948
033	39	1949
034	45	1950
035	46	1951
036	45	1952
037	42	1953
038	34	1954
039	30	1955
040	30	1956
041	31	1957
042	31	1958
043	31	1959
044	32	1960
045	35	1961
046	40	1962
047	42	1963
048	40	1964
049	43	1965
050	47	1966
051	51	1967
052	56	1968

053	53	1969
054	52	1970
055	48	1971
056	46	1972
057	47	1973
058	47	1974
059	39	1975
060	34	1976
061	32	1977
062	33	1978
063	38	1979
064	40	1980
065	38	1981
066	39	1982
067	37	1983
068	37	1984
069	38	1985
070	40	1986

>

On ajoute au fichier "jap WORDS" un indice appelé "sept" (le "Social, Economic, and Political Threat Index de McCann et Stewin, 1990). Les segments sont tous les titres de la revue "Journal of Applied Psychology", de 1917 à 1990. Comme l'indice "sept" n'a été calculé que sur les années 1920 à 1986, il n'y a pas lieu d'ajouter cette variables aux segments 1 à 3, correspondant aux années 1917 à 1919, ni au-delà de l'année 1986.

Le fichier d'options illustre également les noms, avec leurs "paths" en DOS, des fichiers de type "WORDS" créés dans le directory "c:\usr\jap" du disque C. En entrée, le fichier "WORDS" est identifié comme 'c:...\japrhs.wds', c'est-à-dire un fichier "WORDS" strippé, et en sortie, comme 'c:...\japwad.wds', c'est-à-dire, le fichier "WORDS" strippé auquel on a ajouté la variable "sept" avec ses fréquences par segment.

D. Préparation des données

Rien d'autre que ce qui a été décrit dans les paramètres et options, ainsi que dans l'exemple.

E. Autres informations

Aucune.

F. Remarques

Aucune.

G. Messages propres au programme

*CWAD761E	inaccurate alignment
*CWAD762E	event identification syntax error
*CWAD763E	invalid segment number
*CWAD764E	segment number out of range
*CWAD765E	invalid frequency
*CWAD766E	duplicate identification & segment number

XXIII. Le programme CWEDIT

A. But du programme

Le but du programme CWEDIT est d'exécuter les modifications préparées par le programme CFCHECK d'abord, et par le programme CFWKWIC ensuite, c'est-à-dire la modification de certaines occurrences de certains mots en certains points du texte. Le résultat est un texte, sous forme de fichier-système de type "WORDS", contenant moins de mots dont le sens n'est univoque que par le contexte.

B. Exécution

1. Appel du programme

```
... CWEDIT fn1 fn2
```

a. où fn1 est, en entrée, le nom du fichier de filetype "CWEDIT" (qui est d'ailleurs préparé par le programme CFWKWIC); en sortie, ce nom sera celui des fichiers "WORDS", "PUNCHW", ou "SASWORD", si l'un de ces trois types de fichiers était créé.

b. où fn2 est le nom du fichier-système de type "WORDS" contenant le corpus produit et/ou modifié par l'un des programmes CSCUT --avec option "NARR=1"--, CRWSTRIP, ou une exécution antérieure de CWEDIT.

2. Entrées/sorties: Le programme demande deux entrées, qui sont:

a. le fichier de paramètres et options --contenant notamment les occurrences des mots à modifier-- c'est-à-dire un fichier de type "CWEDIT"; à noter que différents types d'items (séparateurs, ponctuation, etc.) peuvent être transformés par CWEDIT.

b. un fichier-système de type "WORDS", le corpus sous une forme "machine".

Les sorties possibles du programme sont, d'abord et surtout, un nouveau fichier (modifié) de type "WORDS", prêt à être analysé par CDWLOOK ou CWSELECT par exemple, ensuite, éventuellement, un fichier de type "PUNCHW" ou son équivalent SAS, c'est-à-dire un fichier de type "SASWORD", l'un et l'autre contenant la liste des mots triés du corpus.

3. Paramètres et options:

FID = L'identification des occurrences et adresses des mots à rechercher; cette identification est extraite du fichier INFIND/FINDS par CFWKWIC (maximum 8 caractères).

FCOM = Le commentaire associé à FID (ci-dessus) (maximum 64 caractères).

FDATE = La date de création du fichier FINDS (maximum 8 caractères).

FTIME = L'heure de création du fichier FINDS (maximum 8 caractères).

TABLE = "TABLE = 1" permet l'impression de la table des mots triés en ordre alphabétique à l'issue des modifications apportées au corpus par le programme. "TABLE = 0", le défaut, inhibe cette impression.

STAT = Par défaut, "STAT = 1", il y a impression des statistiques de tri, c'est-à-dire nombre de mots différents (types), nombre total de mots (occurrences), et rapports type-occurrences --TTR--, en l'état avant et après les modifications apportées par CWEDIT. "STAT = 0" inhibe l'impression de ces statistiques.

PRINT = Si "PRINT = 1", les occurrences des mots à modifier ou à supprimer sont toutes imprimées. Normalement, (le défaut), il n'y a pas impression et "PRINT = 0".

PUNCH = Si "PUNCH = 1", le contenu de la liste imprimée par l'option "TABLE = 1" est en outre "perforée" sous forme d'un fichier de type "PUNCHW" ou "SASWORD". Le défaut, "PUNCH = 0" inhibe la création d'un tel fichier. Pour le format de perforation de ce fichier, on se reportera au point F. "Remarques" du programme CSCUT.

WORD = Par défaut, (WORD = 1"), le programme CWEDIT crée toujours un fichier de type "WORDS". "WORD = 0" inhibe la création de ce fichier.

Les occurrences de mots (ou racines) à remplacer ou modifier sont configurées à raison d'une occurrence par ligne, selon le format suivant:

col 1 à 8	: numéro de séquence du mot en format libre
col 9 à 11	: vierges
col 12 à 31	: ancien mot du corpus aligné à gauche
col 32 à 34	: vierges
col 35 à 54	: nouveau mot aligné à gauche (remplacement) ou vierges (suppression)
col 55 à 57	: vierges
col 58 à 77	: commentaire optionnel
col 78 à 80	: réserve

Si on souhaite ne pas modifier une occurrence particulière, on peut soit supprimer la ligne en question soit la laisser inchangée.

C. Exemples

1. Exemple de procédure d'appel

a. " ... CWEDIT ISIDA1 ISIDA3".

Le programme crée, par "ISIDA1 CWEDIT", un nouveau fichier de type "WORDS" ("ISIDA1 WORDS") à partir de l'ancien fichier "ISIDA3 WORDS".

2. Exemples de fichiers de paramètres et options

a. Contenu du fichier "ISIDA1 CWEDIT" (entre < et >):

```
<
FID='ISIDA  ',
FDATE='07/08/89',
FTIME='15:03:15',
FCOM='                                           ';
    1095  ENTRE                               ENTRE-D
    1279  ENTRE                               ENTRE-D
    2469  ENTRE                               ENTRE-D
    4145  ENTRE                               ENTRE-D
    4951  ENTRE                               ENTRE-D
    5317  ENTRE                               ENTRE-D
    6891  ENTRE                               ENTRE-D
    1883  MANQUE                             MANQUE-N
    5457  MANQUE                             MANQUER
    5558  MANQUE                             MANQUER
    6509  MANQUE                             MANQUER
    4789  PARTI                              PARTI-N
>
```

Les options, telles qu'elles sont pré-arrangées par le programme CFWKWIC qui précède CWEDIT et le prépare. Les occurrences des mots recherchés ont été modifiées manuellement par l'analyste selon le contexte (qui est donné par le programme précédent CFWKWIC).

D. Préparation des données

Aucune.

E. Autres informations

Aucune.

F. Remarques

Aucune.

G. Messages propres au programme CWEDIT

```
*CWED621S  option "NARR = 1" not provided to "CSCUT"
           -  "WORDS" file cannot be processed
*CWED622E  no match with "xxxxxxx"
*CWED623E  duplicate sequence number
*CWED624S  no word to include in output words file
*CWED625E  inaccurate alignment
*CWED626E  invalid sequence number
*CWED627E  sequence number out of range
*CWED628E  word syntax error
*CWED629E  invalid substitution
*CWED633W  no change
```

XXIV. Le programme CWFLOW

I. But du programme

Comme le programme CWNEW, le programme CWFLOW est une mesure du renouvellement du vocabulaire. CWNEW opère la mesure de renouvellement du vocabulaire de segment en segment; CWFLOW opère la mesure de renouvellement du vocabulaire au moyen d'une moyenne mobile calculée en faisant glisser, d'un mot au suivant, une fenêtre d'une longueur déterminée. Par analogie avec un raisonnement fractal¹⁹, on cherche à observer à un niveau microscopique (avec CWFLOW) ce qu'on observe par ailleurs à un niveau plus macroscopique avec CWNEW. Ceci dit, les deux procédures sont indépendantes l'une de l'autre. L'idée de mesurer le profil de renouvellement du vocabulaire a été développée par Youmans (1991, 1994)²⁰.

A. Exécution

1. Entrées/sorties: CWFLOW demande un fichier de paramètres et options en entrée et crée en sortie un fichier listing. Le fichier de paramètres et options identifie les entrées et sorties, soit:

a. **Entrées:** un fichier de type "WORDS" contenant le texte (segmenté au niveau de CSCUT avec l'option "NARR = 1").

b. **Sortie(s):** En sortie, le fichier de paramètres et options identifie le nom du fichier PUNCHV créé par les options PUNCH=1 et SPUNCH=. Le fichier PUNCHV contient les informations commentées en clair dans le listing correspondant.

2. Paramètres et options;

EXAEQUO = Par défaut, "EXAEQUO = 1", le profil de renouvellement du vocabulaire ignore tous les intervalles qui contiennent le même nombre de nouveaux mots. Par exemple, d'une série de "nouveaux mots", 9 9 9 9 12, on ne retient que les valeurs 9 et 12. "EXAEQUO = 0" prend en compte tous les intervalles, avec ou sans répétition des type. La valeur "EXAEQUO = 0" doit être utilisée lorsqu'on veut soumettre le profil de renouvellement du vocabulaire à des analyses de séries temporelles ou d'autres

¹⁹ Butler, D. L. (1991). Simple geometric fractals. Behavior Research Methods, Instruments, & Computers, 23(2), 160-165.

Salvador, L. L. (1992). The labyrinth of intelligence. From natural intelligence to fractal intelligence (by D. Dubois). L'Année Psychologique, 92(4), 596

²⁰ Youmans, G. (1991). A new tool for discourse analysis: The vocabulary-management profile. Language, 67(4), 763-789.

Youmans, G. (1994). The vocabulary-management profile: Two stories by William Faulkner. Empirical Studies of the Arts, 12(2), 113-130.

	analyses qui éliminent ce qui pourrait être dû aux auto-corrélations dans un profil.
<i>NWEND</i> =	Par défaut, "NWEND = 1", le profil de renouvellement du vocabulaire ignore tous les intervalles qui ne se terminent pas par un nouveau mot. "NWEND = 0" prend en compte tous les intervalles, qu'ils se terminent ou non par un nouveau mot. La valeur "NWEND = 0" doit être utilisée lorsqu'on veut soumettre le profil de renouvellement du vocabulaire à des analyses de séries temporelles ou d'autres analyses qui éliminent ce qui pourrait être dû aux auto-corrélations dans un profil.
<i>FIELD</i> =	La fenêtre de la moyenne mobile a une longueur par défaut de 55 mots. Cette valeur, qui doit toujours être un chiffre impair, peut être modifiée par le paramètre FIELD.
<i>SEGNO</i> =	Par défaut, le programme CWFLOW exécute la recherche du profil de renouvellement du vocabulaire pour tout le texte. Si "SEGNO = <u>n</u> ", CWFLOW ne cherche le profil de renouvellement du vocabulaire que pour le segment de texte correspondant à la valeur de <u>n</u> .
<i>PRINT</i> =	"PRINT = 1" imprime le contenu de l'intervalle de la fenêtre mobile pour <u>chaque</u> déplacement de la fenêtre, c'est-à-dire le texte contenu dans chaque déplacement de la fenêtre. Le défaut est "PRINT = 0".
<i>LL</i> =	Ce paramètre gère le nombre de caractères par ligne imprimée. Par défaut, LL est égal à 80; il peut varier de 36 à 132.
<i>STAT</i> =	"STAT = 1" rappelle quelles sont les unités selon lesquelles le texte a été segmenté et fournit les diverses statistiques de tri (par segment, nombre de mots différents, et nombre total de mots). "STAT = 0" inhibe cette opération.
<i>STRIP</i> =	Par défaut, ("STRIP = 1"), le profil de renouvellement du vocabulaire est obtenu à partir du texte lemmatisé (dernière lemmatisation réalisée). Si "STRIP = 0", le profil de renouvellement est obtenu à partir du texte original.
<i>PUNCH</i> =	"PUNCH = 0", le défaut, ne déclenche aucune perforation. "PUNCH = 1" produit un fichier de type PUNCHV composé des fréquences des "types", c'est-à-dire du nombre de mots différents nouveaux dans chaque glissement de la fenêtre.

SPUNCH = crée un nom de fichier et le chemin qui mène à ce fichier (de type PUNCHV). Par exemple,
SPUNCH = 'c:\...\ern\ernwfl.pcv'.
Cette option ne doit pas être utilisée en VM/CMS.

SWORDS = donne au programme le nom du fichier WORDS et le chemin qui y mène. Cette option ne doit pas être utilisée en VM/CMS.

STEMP = 'c:\...*.tmp' (défaut, STEMP = '*.tmp'). Voir le Chapitre III. Description (D.3. Remarques générales concernant les applications de PROTAN sous DOS, UNIX, et Macintosh).

B. Exemples

1. Exemple de fichier d'exécution

Contenu du fichier "cwflow1.bat":

```
call protan.bat cwflow ernwfl.lis ernwfl.wfl
```

2. Exemple de fichier de paramètres et options

Contenu du fichier de paramètres et options "ernwfl.wfl"

```
aequo = 1, nwend = 1, field = 55, segno = 0, print = 1,  
ll = 80, stat = 1, strip = 1, punch = 1,  
SWORDS = 'c:\protan\tests\ern\ernrws.wds',  
SPUNCH = 'c:\protan\tests\ern\ernwfl.pcv',  
STEMP = 'c:\protan\tests\temporaire\*.tmp';
```

Nous traitons ici un texte dont le fichier-système WORDS s'appelle 'c:\...\ernrws.wds'. Ce texte est constitué de 16 poèmes dus à un poète australien, Ern Malley (1993)²¹, qui n'a jamais existé. Ce recueil de poèmes est un faux littéraire notoire dans l'histoire de la littérature australienne (Heyward, 1993)²². Ce qui nous intéresse dans ces poèmes est de savoir si un ordinateur (et PROTAN) peuvent, à deux, faire la distinction entre de vrais poèmes et des faux.

On a fait glisser une fenêtre de 55 mots sur toute la longueur du texte (le nom du fichier WORDS de ce texte est

²¹ Malley, E. (1993). Collected poems. Pymble, Australia: Angus & Robertson.

²² Heyward, M. (1993). The Ern Malley affair (Paperback ed.). Queensland: University of Queensland Press.

'c:\...ernrws.wds') et on supprime tous les intervalles qui contiennent le même nombre de nouveaux mots et tous les intervalles qui ne se terminent pas par un nouveau mot. Le profil est calculé sur le texte strippé. Le profil qui en résulte est présenté à la Figure 1.

On remarque qu'il y a une auto-corrélation forte qui devrait être neutralisée si on souhaitait effectuer sur ce profil une quelconque analyse statistique. On aurait pu réduire la pente de la courbe en éliminant, par une lemmatisation appropriée (via CRWSTRIP), les mots-outils les plus fréquents (du, de, la, le, les, etc.). Mais cette lemmatisation ad hoc n'aurait quand même pas éliminé l'auto-corrélation existante.

De manière générale, lorsqu'on veut effectuer une analyse statistique sur un profil de renouvellement du vocabulaire, il faut mettre les paramètres EXAEQUO et NWEND à la valeur 0, de manière à reconstituer la série complète des mots nouveaux différents pour chaque déplacement de la fenêtre.

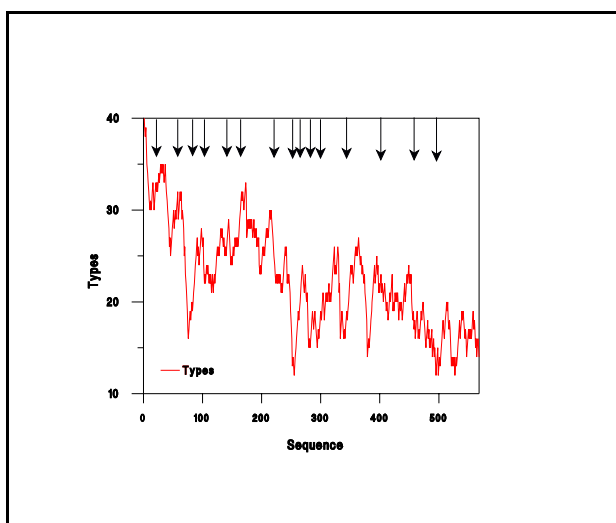


Figure 1. Profil de renouvellement du vocabulaire dans les 16 poèmes d'Ern Malley: Moyenne mobile des nouveaux mots de vocabulaire introduits dans des intervalles successifs de 55 mots de texte. Les flèches indiquent les limites des 16 poèmes.

C. Préparation des données

L'option "NARR = 1" doit avoir été prévue dans le programme CSCUT.

D. Autres informations

Le format de perforation du fichier de type PUNCHV créé par CWFLOW est le suivant:

PUNCHV
créé par CWFLOW
(LRECL = 80)

1er enregistrement:

col 1 à 8	mot-clé "*PUNCHV~"
9 à 16	identification du fichier de sortie
17 à 24	date de création du fichier de sortie
25 à 32	heure de ...(idem)
33 à 40	identification du fichier WORDS en entrée
41 à 48	date de création du fichier WORDS en entrée
49 à 56	heure de ...(idem)
57 à 80	vierges

2me enregistrement:

col 1 à 5	mot-clé "FLOW~"
6 à 10	mot-clé "~POPT"
11 à 15	nombre d'options de traitement fournies à CSCUT
16 à 20	mot-clé "~GCOM"
21 à 25	nombre de lignes de commentaires généraux
26 à 30	mot-clé "~~OPT"
31 à 35	nombre d'options de traitement fournies à CWFLOW
36 à 40	mot-clé "~NSEG"
41 à 45	nombre de segments générés
46 à 80	vierges

3me enregistrement:

col 1 à 64	le commentaire principal du fichier SOURCE
	(celui de l'enregistrement *SOURCE)
65 à 80	vierges

les POPT enregistrements suivants:

col 1 à 80	une option sous la forme XXX=YYY;
------------	-----------------------------------

les GCOM enregistrements suivants:

col 1 à 80	un commentaire général
------------	------------------------

les OPT enregistrements suivants:

col 1 à 80	une option sous la forme XXX=YYY;
------------	-----------------------------------

enregistrements suivants (autant que d'intervalles retenus): chaque variable occupe 8 positions et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

-	nE de séquence moyen de l'intervalle concerné
(variable SEQUENCE)	
-	nombre de nouveaux mots dans cet intervalle
(variable TYPES),	

... et passage à l'enregistrement suivant pour le 2ième intervalle retenu, etc.

E. Remarques

Aucune.

F. Messages propres au programme CWFLOW

```
*CWFL871S  FIELD must be an odd number
*CWFL872S  option "NARR=1" not provided to "CSCUT"
           -  "WORDS" file cannot be processed
*CWFL873S  SEGNO > upper limit in file
*CWFL874S  FIELD > total number of words in "WORDS" file
*CWFL875S  FIELD > total number of words for segment
           SEGNO
```

XXV. Le programme CWKWIC

G. But du programme

Le but du programme CWKWIC est de mettre certains mots du corpus en exergue en présentant ces mots, à partir d'un point fixe de la page, entourés, à gauche et à droite, de leur contexte. L'option "NARR = 1" doit avoir été fournie au programme CSCUT.

H. Exécution

1. Appel du programme

```
... CWKWIC fn1 fn2
```

a. où fn1 est le nom d'un fichier dont le filetype est "CWKWIC", et

b. où fn2 est le nom du fichier-système de filetype "WORDS" --avec option NARR = 1-- créé par CSCUT, CRWSTRIP, CWADD, etc.

2. Entrées/sorties:

CWKWIC ne crée aucun fichier en sortie, sinon une liste imprimée du contexte des mots à localiser (à raison de 40 caractères à gauche et de 40 caractères à droite, mot-cible compris, du mot dont on veut voir le contexte). En entrée cependant, le programme CWKWIC demande un fichier de type "CWKWIC" contenant les paramètres et options et le fichier-système de type "WORDS". Les mots à localiser figurent dans le fichier de type "CWKWIC" après les options. Ces mots doivent être entourés, de part et d'autre, d'une apostrophe et doivent être séparés par une virgule et/ou un ou plusieurs blancs (~). Enfin, le nombre de mots à localiser est limité à MAXREF = 1.000 par défaut, un nombre largement suffisant pour la plupart des applications de recherche de "key-word-in-context".

3. Paramètres et options:

SORTW = Les mots à localiser ne doivent pas être rangés alphabétiquement ("SORTW = 0;", défaut); on peut obtenir un KWIC sur les mots rangés en ordre alphabétique par l'option "SORTW = 1;".

CONTEXT = Les mots à localiser apparaissent avec leur contexte (CONTEXT = 1). Dans le cas contraire (CONTEXT = 0, valeur par défaut), seul est fourni un comptage des fréquences d'apparition par mot-clé.

MAXREF = Le nombre maximum de mots-clés est de 1.000 par défaut, sinon il convient de préciser la valeur de MAXREF.

STAT = Par défaut, "STAT = 0", il n'y a pas impression des statistiques de tri qui accompagnent les fichiers de type "WORDS". "STAT = 1" donne les statistiques du nom-

bre de mots différents et du nombre total de mots, par segment et pour le total.

STRIP = L'opération de "key-word-in-context" peut se faire à partir d'un fichier WORDS strippé (dans ce cas, *STRIP* = 1), ou à partir d'un fichier WORDS non-strippé (dans ce cas, *STRIP* = 0, et c'est la valeur par défaut).

SWORDS = Cette option est nécessaire pour l'utilisation de PROTAN en PC, pour fournir au programme le nom (et le chemin éventuellement) du fichier WORDS. Par exemple, *SWORDS* = 'd:....CAMUSrws.wds'.

I. Exemples

1. Exemples de procédure d'appel

a. " ... CWKWIC PSYCHO NEURO".

Le fichier "PSYCHO CWKWIC" contient les mots à localiser et le fichier "PSYCHO WORDS" est le fichier-système.

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "PSYCHO CWKWIC" (entre < et >):

```
<
CONTEXT = 1, STRIP = 0;
'JAMAIS' 'MAIS' 'MAIS'
>
```

On ne souhaite pas faire ranger en ordre alphabétique les mots à localiser. L'option "SORTW = 0;" est donc prise par défaut. Suivent 3 mots-cibles, 'JAMAIS', 'MAIS', et '~MAIS'.

J. Préparation des données

L'option "NARR = 1" doit avoir été fournie antérieurement lors de l'exécution du programme CSCUT. Pour le reste, veiller à entourer les mots à localiser d'une apostrophe de part et d'autre, séparés par une virgule et/ou un ou plusieurs blancs (~).

On notera que les résultats du KWIC diffèrent selon que les mots-cibles sont, ou ne sont pas, précédés et/ou suivis d'un blanc. Par exemple, le mot-cible "MAIS" va "prendre" dans un texte les mots 'mais', 'jamais', et 'maison', alors que le mot-cible "~MAIS" (cette fois précédé d'un blanc) ne prend que 'mais' et 'maison'; par contre, le mot-cible "~MAIS~", précédé et suivi d'un blanc, ne prendrait que les occurrences du mot 'mais'.

K. Autres informations

Le résultat du dernier exemple ci-dessus figure dans l'encadré qui suit²³. (Pour une raison de mise en page, on n'a pas pu reproduire toutes les informations présentes en amont sur la liste imprimée; et pour la même raison, il n'a pas été possible de faire apparaître toute la largeur des lignes imprimées, les lignes du manuel étant limitées à 72 caractères).

```
PL=60;  
SLISTING='camuswki.lis';  
SMASTER='camuswki.wki';
```

```
**          PROTAN SYSTEM          UCL/PSP AT LOUVAIN-LA-NEUVE    PAGE:      1  **  
**          PROGRAM: CWKWIC        DATE: Nov01 91    TIME: 13:07:00    **  
  
OPTIONS IN EFFECT _____  
  
CONTEXT=1;  
MAXREF=1000;  
SORTW=0;  
STAT=0;  
STEMP='*.tmp';  
STRIP=0;  
SWORDS='camuswad.wds';  
*CWKI032I  end of data set reached on "camuswki.wki"  
- parameter list complete
```

²³ Tiré de "La Chute" de Camus.

```

**          PROTAN SYSTEM                UCL/PSP AT LOUVAIN-LA-NEUVE          PAGE:      2      **
**          PROGRAM: CWKWIC              DATE: Nov01 91          TIME: 13:07:00      **

"WORDS" FILE DEFINITIONS _____

"WORDS" FILE IDENTIFICATION (USER-DEFINED)..... CAMUS
"WORDS" FILE CREATION DATE..... Nov01 91
"WORDS" FILE CREATION TIME..... 13:04:51
USER'S COMMENT.
INPUT "WORDS" FILE PROCESSING LEVEL.. 2

PROCESSING OPTIONS PROVIDED TO "CSCUT" _____

BRKB='I';
BRKM=1;
CTRL=200;
NARR=1;
SEGT='W';
SENT='.!.??:';

NUMBER OF SEGMENTS GENERATED : 14 _____

GENERAL COMMENTS (HISTORY OF "SOURCE" & "WORDS" FILES AND OTHER COMMENTS) _____

PROC BY CWADD      "CAMUS"      " (Nov01 91 13:04:51)
PROC BY CWEDIT     "CAMUS"      " (Nov01 91 12:05:43) "CAMUS"      " (Nov01 91 11:31:18)
PROC BY CRWSTRIP   "CAMUS"      " (Nov01 91 10:58:02) "FRSTRP01" (Nov01 91 09:47:55)
PROC BY CSCUT      "CAMUS"      " (Nov01 91 09:34:07)
CAMUS LA CHUTE

```

```

**          PROTAN SYSTEM                UCL/PSP AT LOUVAIN-LA-NEUVE      PAGE:    3   **
**          PROGRAM: CWKWIC              DATE: Nov01 91        TIME: 13:07:11     **
** CAMUS                                **                               **


          OUTPUT SORT STATISTICS


KEYWORD                      TOTAL
                              WORDS _____
' JAMAIS '                   1
'MONSIEUR'                   7
TOTAL                        8

```

```
**      PROTAN SYSTEM      UCL/PSP AT LOUVAIN-LA-NEUVE      PAGE:      4      **
**      PROGRAM: CWKWIC      DATE: Nov01 91      TIME: 13:07:11      **
** CAMUS**
```

KEYWORD IN CONTEXT

SEG NO	SEQ NO	SEG FREQ	TOT FREQ
--------	--------	----------	----------

*** KEYWORD ' JAMAIS '

S QUITTE PRE2S DE CE PONT . JE NE PASSE JAMAIS SUR UN PONT , LA NUIT . C EST LA

14	3170	1	1
----	------	---	---

*** KEYWORD 'MONSIEUR'

PUIS-JE , MONSIEUR , VOUS PROPOSER MES SERVICES ,

1	3	1	1
---	---	---	---

ES GRANDS ANIMAUX . MAIS JE ME RETIRE , MONSIEUR , HEUREUX DE VOUS AVOIR OBLIGE1

1	146	2	2
---	-----	---	---

R . N INSISTONS PAS , MAIS CROYEZ MOI , MONSIEUR , TOUTES LES SURPRISES SONT POS

10	2223	1	3
----	------	---	---

VERTES . LA N-PLHOLLANDE EST UN SONGE , MONSIEUR , UN SONGE D OR ET DE FUME1E ,

12	2686	1	4
----	------	---	---

E PLAIDE . PARDONNEZ MOI . L HABITUDE , MONSIEUR , LA VOCATION , LE DE1SIR AUSSI

13	2900	1	5
----	------	---	---

LA2 , JE LES ATTENDS . A2 DEMAIN DONC , MONSIEUR ET CHER COMPATRIOTE . NON , VOUS

14	3146	1	6
----	------	---	---

S , DERRIERE CES VITRINES ? LE RE3VE , MONSIEUR , LE RE3VE A2 PEU DE FRAIS , LE

14	3256	2	7
----	------	---	---

```
**      PROTAN SYSTEM      UCL/PSP AT LOUVAIN-LA-NEUVE      PAGE:      5      **
**      PROGRAM: CWKWIC      DATE: Nov01 91      TIME: 13:07:11      **
```

```
*CYEX001I processing terminated
- normal end of job
```

L. Remarques

Aucune.

M. Messages propres au programme CWKWIC

```
*CWKI781S option "NARR=1" not provided to "CSCUT"
- "WORDS" file cannot be processed
```

XXVI. Le programme CWKWOC

A. But du programme

Le but de CWKWOC est de mettre en évidence, sous forme de liste, un ensemble de mots du corpus à partir d'un ensemble de mots-clés constitués de mots ou parties de mots. En mettant ainsi des mots "hors contexte", on se donne la possibilité de créer, avec un minimum de manipulations, de nouvelles entrées dans les dictionnaires de formes nominales (existants ou à élaborer). Alternativement, des listes de mots en format KWOC peuvent être utilisées, sans autres modifications, comme fichiers de référence "REFER" par les programmes CWSELECT et CWORD. CWKWOC produit aussi des listes de mots en format "INFIND" qui peuvent être utilisées telles quelles comme fichier de recherche de mots "INFIND" par le programme CFCHECK.

B. Exécution

1. Appel du programme

```
... CWKWOC fn1 fn2
```

a. où fn1 est le nom d'entrée (filename) d'un fichier de type CWKWOC; en sortie, ce nom sera celui de fichiers de type "PUNCHW" (ou "SASWORD"), "INROOT", "REFER" et "INFIND", prêts à être utilisés soit par le programme CFCHECK, dans un fichier de formes nominales de type INROOT, soit par les programmes CWSELECT et CWORD, dans un fichier de référence de type REFER, soit par le programme CFCHECK dans un fichier de recherche de mots de type "INFIND";

b. où fn2 est le nom du fichier-système de filetype "WORDS", créé précédemment par CSCUT, CRWSTRIP, ou CWEDIT.

2. Entrées/sorties: Le programme CWKWOC demande en entrée un fichier de type CWKWOC ("fn1 CWKWOC") contenant les paramètres et options, ainsi que les mots-clés destinés à mettre des mots du corpus "hors contexte". Ces mots-clés doivent être entourés d'une apostrophe de part et d'autre, et séparés par une virgule et/ou au moins un blanc (~); le nombre de mots-clés est limité à 1.000.

Le programme demande également un fichier "fn2 WORDS" qui est le corpus sous forme de fichier-système (créé par CSCUT, CRWSTRIP ou CWEDIT): Dans ce fichier-système se trouvent les mots qu'on veut mettre en évidence en les mettant dans une liste distincte.

Comme première sortie, le programme crée, en option, une liste imprimée des mots (ou mots contenant certaines chaînes de caractères) qu'on veut voir "hors contexte". L'option PUNCH = 0, 1, 2, ou 3 est la seconde sortie créée par le programme CWKWOC.

3. Paramètres et options:

SORTW = Les mots-clés peuvent être rangés en ordre alphabétique à l'impression ("SORTW = 1"); le travail de mise en évidence sera alors effectué selon cet ordre. Par défaut

("SORTW = 0"), le travail de mise en évidence s'effectue selon l'ordre existant dans le fichier "fn1 CWKWOC".

STAT = Par défaut, "STAT = 0", il n'y a pas impression des statistiques de tri qui accompagnent les fichiers de type "WORDS". "STAT = 1" donne les statistiques du nombre de mots différents et du nombre total de mots, par segment et pour le total.

TABLE = La table des mots "hors contexte" (liste alphabétique par mot-clé) sera imprimée si TABLE vaut 1. Si "TABLE = 0", le défaut, la table n'est pas imprimée. Cette table indique, pour chaque mot mis hors contexte, le numéro du segment où le mot se présente, sa fréquence dans ce segment, et, dans le dernier segment d'occurrence, sa fréquence totale dans l'ensemble des segments. Chaque mot-clé donne lieu à l'impression d'une table.

PUNCH = "PUNCH = 0", le défaut, inhibe toute perforation de données.

"PUNCH = 1" perfore, pour chaque mot mis "hors contexte", le numéro du segment où le mot apparaît, sa fréquence dans ce segment, et, dans le dernier segment d'occurrence, sa fréquence totale dans tous les segments. Le format de "PUNCH = 1" correspond à celui d'un fichier de type "PUNCHW" (voir le programme CSCUT).

"PUNCH = 2" perfore la liste alphabétique des mots différents correspondant, dans le fichier WORDS, aux divers mots-clés. Mais ici, la liste des mots qui apparaît dans les colonnes 1 à 20 est redoublée dans les colonnes 24 à 43. En colonnes 47 à 53 ne figure plus que la fréquence totale de chaque mot. Le format des données perforées par le paramètre "PUNCH = 2" est en réalité celui d'un fichier de type "INROOT", et c'est bien à cela que doit servir cette opération. (Voir plus loin, "Autres informations").

"PUNCH = 3" perfore encore la liste alphabétique des mots mis "hors contexte" par le programme. Seuls les mots différents figurent encore dans ce fichier "perforé", entourés en outre, de part et d'autre, d'une apostrophe. Aucune indication de segment ni de fréquence n'y figure. Le format des données perforées par le paramètre "PUNCH = 3" est celui d'un fichier de type "REFER", immédiatement utilisable par les programmes CWSELECT et CWORD.

"PUNCH = 4" enfin perfore la liste alphabétique des mots différents mis "hors contexte" par le programme. Aucune apostrophe ni indication de segment ou de fréquence n'y figure et le format du fichier produit est celui d'un fichier de type "INFIND", utilisable tel quel par le programme CFCHECK en vue de la création d'un fichier-système de type "FINDS".

Notons encore que les options PUNCH et TABLE sont indépendantes l'une de l'autre.

C. Exemples

1. Exemple de procédure d'appel

a. " ... CWKWOC STRP02 INTRUSE".

Le fichier "STRP02 CWKWOC" contient les paramètres et options ainsi que les mots-clés. Le fichier "INTRUSE WORDS" est le corpus, sous forme de fichier-système, où se trouvent les mots qui seront mis en évidence.

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "STRP02 CWKWOC" (entre < et >):

```
<
TABLE = 0, STAT = 0, PUNCH = 2;
'AIENT ' 'AIT '
>
```

On désire seulement obtenir, sous forme de résultat perforé, la liste des mots différents qui, dans le corpus, se terminent par les chaînes de caractères "AIENT" et "AIT"; le résultat sera en fait une liste composée principalement des formes conjuguées des verbes. (Voir le résultat de l'exécution dans "Autres informations").

D. Préparation des données

Aucune, sinon à entourer les mots-clés d'une apostrophe de part et d'autre, séparés par une virgule et un ou plusieurs blancs (~).

Par ailleurs, on retiendra que l'effet d'un mot-clé diffère selon l'absence de blanc (~) ou la présence de blanc(s) (~) devant et/ou derrière le mot-clé. Schématiquement, cet effet peut être résumé comme suit:

```
mot-clé 'xxx'..... met hors contexte toutes les occurrences
                        de xxx

mot-clé '~xxx'..... met hors contexte tous les mots commençant
                        par xxx

mot-clé 'xxx~'..... met hors contexte tous les mots se
                        terminant par xxx

mot-clé '~xxx~'.... met hors contexte les seules occurrences
                        du mot xxx
```

E. Autres informations

A partir des fichiers "INTRUSE WORDS" et "STRP02 CWKWOC" (voir plus haut, "Exemple de fichier de paramètres et options"), on obtient le fichier de type INROOT suivant:

```
*INROOT-INTRUSE
PROC BY CWKWOC "INTRUSE " (24/03/88 15:24:46)
PROC BY CRWSTRIP "INTRUSE " (24/03/88 15:24:29)
"FRSTRP01" (14/03/88 10:27:40)
PROC BY CSCUT "INTRUSE " (24/03/88 15:24:04)
L'INTRUSE, DE J. L. BORGES, IN LE RAPPORT DE BRODIE
*ENDCOM
ACCOMPAGNAIT ACCOMPAGNAIT 1
ATTELAIT ATTELAIT 1
CE1DAIT CE1DAIT 1
CHERCHAIENT CHERCHAIENT 1
CIRCULAIT CIRCULAIT 1
COMBLAIT COMBLAIT 1
CONFIRMAIT CONFIRMAIT 1
CRAIGNAIT CRAIGNAIT 1
DE1FENDAIENT DE1FENDAIENT 1
DIFFE1RAIENT DIFFE1RAIENT 1
DISPUTAIENT DISPUTAIENT 1
ENIVRAIT ENIVRAIT 1
EXHIBAIT EXHIBAIT 1
GAGNAIT GAGNAIT 1
HAUSSAIT HAUSSAIT 1
HUMILIAIT HUMILIAIT 1
IGNORAIT IGNORAIT 1
JALOUSAIENT JALOUSAIENT 1
PRONONCAIENT PRONONCAIENT 1
RENDAIENT RENDAIENT 1
RO3DAIT RO3DAIT 1
SCANDALISAIT SCANDALISAIT 1
SOUCIAIT SOUCIAIT 1
SOUVENAIT SOUVENAIT 1
SUFFISAIT SUFFISAIT 1
TAISAIT TAISAIT 1
UNISSAIT UNISSAIT 1
```

Il s'agit de certaines formes conjuguées de verbes contenus dans le fichier-système "INTRUSE WORDS". Il suffit alors de changer la forme conjuguée dans la colonne de droite en la forme nominale de ce verbe. On obtient alors:

```

*INROOT-STRP02
PROC BY CWKWOC      "INTRUSE " (24/03/88 15:24:46)
PROC BY CRWSTRIP    "INTRUSE " (24/03/88 15:24:29)
                   "FRSTRP01" (14/03/88 10:27:40)
PROC BY CSCUT       "INTRUSE " (24/03/88 15:24:04)
L'INTRUSE, DE J. L. BORGES, IN LE RAPPORT DE BRODIE
*ENDCOM
ACCOMPAGNAIT        ACCOMPAGNER                1
ATTELAIT            ATTELER                      1
CE1DAIT             CE1DER                      1
CHERCHAIENT         CHERCHER                    1
CIRCULAIT           CIRCULER                    1
COMBLAIT            COMBLER                     1
CONFIRMAIT          CONFIRMER                   1
CRAIGNAIT           CRAINDRE                    1
DE1FENDAIENT        DE1FENDRE                   1
DIFFE1RAIENT        DIFFE1RER                  1
DISPUTAIENT         DISPUTER                    1
ENIVRAIT            ENIVRER                     1
EXHIBAIT            EXHIBER                     1
GAGNAIT             GAGNER                      1
HAUSSAIT            HAUSSE                      1
HUMILIAIT           HUMILIER                   1
IGNORAIT            IGNORER                     1
JALOUSAIENT         JALOUSE                    1
PRONONCAIENT        PRONONCER                  1
RENDAIENT           RENDRE                     1
RO3DAIT             RO3DER                     1
SCANDALISAIT        SCANDALISER                1
SOUCIAIT            SOUCIER                    1
SOUVENAIT           SOUVENIR                   1
SUFFISAIT           SUFFIRE                    1
TAISAIT             TAIRE                      1
UNISSAIT            UNIR                       1

```

Les fréquences figurant en colonne 47 à 53 sont prises par le programme CRCHECK comme commentaires: On peut les y laisser ou les effacer. La dernière opération consiste alors à utiliser ce fichier dans un programme CRCHECK pour créer un fichier-système de type 'ROOTS'. Il n'y a plus alors qu'à lancer la procédure d'appel

```
< ... CRWSTRIP fn1 STRP02 INTRUSE>
```

pour obtenir un fichier-système ("fn1 WORDS") dépouillé des formes conjuguées ('AIT' et 'AIENT') des verbes ("fn1 CRWSTRIP" est le fichier de paramètres et options de CRWSTRIP, voir ce programme).

Alternativement, on peut inclure ce fichier 'STRP02 INROOT' en supplément d'un dictionnaire de formes nominales existant (à condition d'en effacer les premiers enregistrements jusqu'à "*ENDCOM" compris).

F. Remarques

Aucune.

G. Messages propres au programme CWKWOC

Aucun

XXVII. Le programme CWNEW

A. But du programme

CWNEW a pour objectif d'identifier et compter les mots qui apparaissent pour la première fois dans un texte (par rapport à un ou une série de textes antérieurs).

B. Exécution

1. Appel du programme

```
... CWNEW fn1 fn2 [fn3...]
```

a. où fn1 est le nom d'un fichier de filetype "CWNEW",

b. où fn2, fn3, ... sont les noms des fichiers (de filetype "WORDS") qui contiennent les textes ou parties de textes qui vont être comparés entre eux.

2. Entrées/sorties: CWNEW ne crée pas de fichier-système en sortie; par contre, CWNEW crée une liste imprimée ainsi qu'un fichier PUNCH.

En entrée, CWNEW demande deux types de fichiers. Un fichier "fn1 CWNEW" d'abord, qui contient les paramètres et les options, et ensuite un ou plusieurs fichiers "fn2 WORDS", "fn3 WORDS", etc. (jusque 9 fichiers), contenant les textes dont on veut connaître les caractéristiques d'accroissement du vocabulaire.

3. Paramètres et options:

STAT = Par défaut, "STAT = 0", il n'y a pas impression des statistiques de tri qui accompagnent les fichiers de type "WORDS". "STAT = 1" donne les statistiques du nombre de mots différents et du nombre total de mots pour les segments demandés; "STAT = 1" donne également, sur la même ligne, des indications sur la nature "old ou new" du segment.

TABLE = Lorsque "TABLE = 1", la liste des mots qui apparaissent pour la première fois dans le texte est imprimée. "TABLE = 0" inhibe cette impression.

PUNCH = Lorsque "PUNCH = 1", il y a création d'un fichier de type "PUNCHI" ou "SASINC" ("--I" et "--INC" pour "INCrease") contenant les indices d'accroissement du vocabulaire des textes soumis à l'analyse. Ces indices, répétés pour chaque segment, se répartissent sur 7 colonnes qui sont, de gauche à droite:

- le numéro du traitement
- le nombre de mots différents "anciens", c'est-à-dire apparaissant dans le ou les textes situés en amont,
- le nombre total de mots "anciens" (en amont),

- le nombre de mots différents "nouveaux", c'est-à-dire apparaissant dans le ou les textes situés en aval,
- le nombre total de mots "nouveaux" (en aval),
- le nombre de mots différents dans les textes "en aval" qui n'étaient pas encore apparus dans les textes situés en amont,
- et le nombre total de mots dans les textes "en aval" qui n'étaient pas encore apparus dans les textes situés en amont.

Il n'y a pas de fichier "PUNCH" si l'option PUNCH vaut 0.

WORDNT = On peut faire imprimer la liste des mots qui ne figurent pas parmi ceux qui sont repris par l'option "TABLE = 1", mais qui auraient pu l'être eu égard à leurs n premiers caractères, n étant la valeur qu'on donne à WORDNT, par exemple, "WORDNT = 5". "WORDNT = 0", le défaut, inhibe cette option.

La notion d'accroissement du vocabulaire est en effet à la fois relative et continue, chaque segment de texte pouvant être successivement: (1) un texte dont on veut connaître les mots qui apparaissent pour la première fois (par rapport à un ou à des textes situés en amont), ou (2) un texte dont on veut connaître les mots qui sont déjà apparus au moins une fois (par rapport à un ou à des textes situés en aval).

On va donc avoir besoin, alternativement, (1) de définitions de segment(s) (en format libre), chaque définition ne pouvant occuper qu'une ligne, et (2) d'un mot-clé qui déclenche le traitement. Les deux parties de ces paramètres sont répétées autant de fois qu'il y a de textes à "comparer".

(1) La définition de segment en format libre se fait de la manière suivante:

- identification du fichier "WORDS" (voir "*SOURCE-i-identification)
- numéro du segment (éventuellement en série, c'est-à-dire 2 numéros de segments séparés par un tiret --avec ou sans blanc (~))
- l'appartenance du ou des segments au groupe de mots anciens (OLD) ou nouveaux (NEW), indiquée respectivement par les lettres O (pour OLD), et N (pour NEW)
- commentaire éventuel.

Les 4 éléments de cette définition de segments doivent toujours être séparés par au moins un blanc (~).

(2) L'ordre d'exécution du traitement (en fonction des définitions de segments) se fait de la manière suivante:

- un caractère "/"
- le mot-clé "RUN"
- un commentaire éventuel, celui-ci étant cette fois utilisé comme titre de page pour le traitement correspondant. Un blanc (~) précède ce commentaire.

C. Exemples

1. Exemple de procédure d'appel

a. " ... CWNEW FCCA FCCB".

On cherche à connaître les caractéristiques d'accroissement du vocabulaire (paramètres et options définis dans "FCCA CWNEW") d'un fichier "FCCB WORDS".

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "FCCA CWNEW" (entre < et >):

```
<
TABLE=1, STAT=1, PUNCH=1;
CCC 1      O
CCC 2      N
/RUN
CCC 1-2    O
CCC 3      N
/RUN
CCC 1-3    O
CCC 4      N
/RUN
CCC 1-4    O
CCC 5      N
/RUN
CCC 1-5    O
CCC 6      N
/RUN
>
```

On cherche ici à connaître, d'un texte au suivant, les caractéristiques d'accroissement du vocabulaire d'une suite de 6 textes.

D. Préparation des données

La préparation des données consiste surtout à combiner, deux à deux, les séquences de textes de manière à ce que les caractéristiques d'accroissement du vocabulaire puissent être reliées à d'autres informations sur la permanence d'un style ou, au contraire, sur des modifications de style ou de thèmes ou encore à des changements d'auteurs.

E. Autres informations

A partir du fichier "FCCA CWNEW" décrit ci-dessus, on obtient, outre une liste imprimée, un fichier PUNCH qui contient les divers indices d'accroissement du vocabulaire (décrits dans l'option "PUNCH ="). Contenu du fichier "FCCA PUNCHI":

*PUNCHI						
27/06/8915:50:25						
INC	NWDS	1				
	CCC	27/06/8915:49:55				
1	656	1950	680	1890	444	543
2	1100	3840	785	2603	444	601
3	1544	6443	1025	3542	532	679
4	2076	9985	966	3384	426	494
5	2502	13369	662	2178	240	283
6	2742	15547	814	3040	256	299

Il est apparent, déjà dans cet exemple, que dans des textes provenant d'une même source, les rapports du nombre de mots apparaissant pour la première fois au nombre total de mots nouveaux, c'est-à-dire les rapports des colonnes 7 à 5, tendent à décroître progressivement. On observe en effet:

col 7/ col 5	
-	-
543 / 1890	= .29
601 / 2603	= .23
679 / 3542	= .19
494 / 3384	= .15
283 / 2178	= .13
299 / 3040	= .10

F. Remarques

Le fichier perforé que crée l'option "PUNCH = 1" ("PUNCHI" ou "SASINC") est destiné à être traité par l'un ou l'autre programme d'analyse statistique. Ce fichier perforé se présente comme suit:

PUNCHI
créé par CWNEW
(LRECL = 80)

1er enregistrement:

col 1 à 8 mot-clé "*PUNCHI~"
9 à 16 vides
17 à 24 date de création du fichier de sortie
25 à 32 heure de ...(idem)
33 à 80 vides

1me enregistrement:

col 1 à 5 mot-clé "INC~~"
6 à 10 mot-clé "~NWDS"
11 à 15 nombre de fichiers WORDS définis pour le traitement
16 à 80 vides

les NWDS enregistrements suivants:

col 1 à 8 vides
9 à 16 identification d'un fichier WORDS
17 à 24 date de création de ce fichier WORDS
25 à 32 heure de ...(idem)
33 à 80 vides

enregistrements suivants: chaque variable occupe 8 positions et est séparée de la suivante par un caractère de tabulation. Soit, pour chaque enregistrement:

- numéro d'ordre du traitement (variable PROC)
- nombre de mots différents "anciens" (variable WDO)

WDN) - nombre total de mots "anciens" (variable WTD)
 - nombre de mots différents "nouveaux" (variable
groupe des "anciens" - nombre total de mots "nouveaux" (variable WTN)
 - nombre de mots différents "nouveaux" absents du
 (variable WDA)
des "anciens" - nombre total de mots "nouveaux" absents du groupe
 (variable WTA)

... et passage à l'enregistrement suivant pour le traitement suivant.

G. Messages propres au programme CWNEW

*CWNE721S	keyword "RUN" missing
*CWNE722S	keyword "RUN" syntax error
*CWNE723S	segment definition missing
*CWNE724S	seg2 less than seg1
*CWNE725S	Old/New missing
*CWNE726S	Old/New syntax error
*CWNE727S	segment value > upper limit in file
*CWNE728S	identification length > 8
*CWNE729S	invalid char. in segment value
*CWNE730S	number of Old/New segments must be both > 0
*CWNE732S	duplicate segment definition for file "xxxxxxx"
	-- segment nnnnn
*CWNE736S	file identification not matched
*CWNE737S	duplicate file identification

XXVIII. Le programme CWPAT

A. But du programme

Il est dans la nature du langage, et de l'être humain, d'utiliser des expressions toutes faites. Il s'agit de ces éléments stéréotypés du langage qu'on utilise pour leur pouvoir évocateur, même si leur pouvoir mobilisateur est perdu depuis longtemps. Le programme CWPAT de reconnaissance de patterns va aider l'analyste à localiser et identifier ces "expressions toutes faites" à partir de quelques éléments de ces patterns que l'analyste fournit au programme.

B. Exécution

1. Appel du programme

```
... CWPAT fn1 fn2
```

a. où fn1 est le nom du fichier de type CWPAT qui contient les options et paramètres nécessaires pour reconnaître les patterns de mots.

b. où fn2 est le nom du fichier de type "WORDS", c'est-à-dire le texte dans lequel on va rechercher les diverses configurations de mots. *Ce fichier "WORDS" doit avoir été créé avec l'option "NARR = 1" par le programme CSCUT.*

2. Entrées/sorties: CWPAT a deux entrées et une seule sortie, soit, en entrée, le fichier de paramètres/options et le texte sous forme de fichier "WORDS", et en sortie, une liste imprimée contenant les diverses configurations de recherche.

3. Paramètres et options:

STAT = "STAT = 1" récapitule les définitions de segment demandées par l'utilisateur et donne les statistiques de tri, c'est-à-dire, par segment demandé, le nombre de mots différents et le nombre total de mots. "STAT = 0" (le défaut) inhibe cette impression.

STRIP = Par défaut, ("STRIP=0"), la recherche de patterns se fait sur un texte non lemmatisé. Si "STRIP = 1", la recherche de patterns se fait sur le niveau de lemmatisation le plus récent.

CONTEXT = Par défaut, PROTAN rapporte le résultat d'une recherche de configuration en signalant le nombre de patterns pertinents trouvés. Si "CONTEXT = 1", les mots paramétrés vont en outre être présentés dans leur contexte, dans les limites de la longueur du champ (voir "FIELD =").

MAXREF = concerne le nombre maximum de mots-clés qu'on peut mettre dans une recherche. Par défaut, 20 mots est le nombre de mots-clés

acceptés dans chaque requête. Plus le nombre de mots-clés est élevé, moins il y a de chances de trouver des configurations contenant exactement ces mots-clés, et encore moins si on les demande dans un certain ordre.

- MAXTAG** = Par défaut, le nombre total de mots trouvés pour un pattern est de 1.000: c'est la valeur par défaut de MAXTAG. Lorsqu'on est en présence d'un corpus volumineux, on devra parfois accroître la valeur de MAXTAG à 5.000 ou 10.000 (peu de mémoire requise pour ce changement de paramètre). Par exemple, si on exploite un pattern susceptible d'atteindre une fréquence élevée, par exemple dans un corpus de 100.000 mots, il est prudent de modifier la valeur de MAXTAG en fixant pour MAXREF une valeur "préventive" de 5.000 ou 10.000.
- FIELD** = Par défaut, on fait courir une "fenêtre" de recherche jusqu'à la limite du segment. Ce défaut s'écrit "FIELD = -1" (1 pouvant être aussi n'importe quel autre nombre négatif). Par contre, si FIELD est un nombre positif ("FIELD = 12" par exemple), l'étendue de la "fenêtre" de recherche est limitée à la valeur de "FIELD =".
- EXTEND** = Il est utile de préciser si la prochaine recherche de patterns (à l'intérieur d'une requête) doit commencer avec le dernier mot trouvé ("EXTEND = 0", défaut) ou bien juste après le dernier mot trouvé ("EXTEND = 1").
- TEXTW** = Si "TEXTW = 1" (défaut), les différentes occurrences des mots paramétrés, avec les numéros de séquence et de segment, sont reproduites. Sinon ("TEXTW = 0"), cette étape est absente de la liste imprimée.
- SWORDS** = En DOS, il faudra donner le nom du fichier "WORDS" précédé de son "path", par exemple, 'd:\protan\tests\hlm\hlmscu.wds'.

Les différentes requêtes figurent après la dernière option: une série de mots-clés organisés selon une certaine syntaxe, chaque requête (à commencer toujours à la ligne) se termine par le signe "/" (pouvant être suivi d'un commentaire).

Syntaxe des paramètres de CWPAT.

Deux ou plusieurs mots juxtaposés sont implicitement reliés par une opération logique "ET". Par exemple:

cercle enfer / première requête

est la syntaxe de recherche de mots reliés par "ET".

Par contre, des mots situés à l'intérieur d'une paire de parenthèses sont reliés par l'opération logique "OU". Par exemple: (cercle rond) enfer / deuxième requête permet de connaître les endroits où "cercle" ou "rond" sont précédés ou suivis du mot "enfer".

Enfin, les mots situés à l'intérieur d'une paire de crochets sont reliés par l'opération logique "SUIVI DE". Par exemple: <cercle enfer> / troisième requête va relever tous les cas où le mot "cercle" est suivi du mot "enfer". *A noter que l'opération logique "SUIVI DE" ne peut pas être incluse elle-même dans une opération logique "OU".*

Notons que les mots-clés peuvent être remplacés par leurs racines, par exemple "rond.", pouvant prendre la place de "ronde", "rondeur", "rondins", "rondelle", etc.

C. Exemples de procédure d'appel

1. Exemple de procédure d'appel

a. " ... CWPAT patterns JAP".

Le fichier "patterns CWPAT" contient les options et paramètres de recherche de configuration dans le texte "JAP WORDS".

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "patterns CWPAT" (entre < et >);

```
<
swords='hlmrws.wds',
EXTEND=1,TEXTW=1, MAXREF=100,MAXTAG=500,
CONTEXT=1,STAT=1,STRIP=0,FIELD=20;
north (crass. lack. ) /
<south (crass. lack)>    /
(north south) lack.    /
>
```

On recherche un certain nombre de patterns à l'intérieur d'une fenêtre de 20 et on désire connaître dans quels contextes ces patterns apparaissent. Voici le résultat de cette recherche (sur un texte du journaliste polémiste américain H. L. Mencken (1880-1956)):

```
**      PROTAN SYSTEM      UCL/PSP AT LOUVAIN-LA-NEUVE      PAGE:      3      **
**      PROGRAM: CWPAT      DATE: Mar11 93      TIME: 21:58:44      **
** HLM
```

north (crass. lack.)/

TEXT WORD	SEQ NO	SEG NO
CRASS	26	1
NORTH	53	1
CRASSNESS	63	1
NORTH	68	2
NORTH	84	2
LACK	94	2

SEG NO : 1
SEQ NO : 53

NORTH , OF COURSE , THERE IS ALSO GROSSNESS , CRASSNESS

3 4 5 6 7 8 9 0 1 2 3

SEG NO : 2
SEQ NO : 84

NORTH IS THERE SUCH COMPLETE STERILITY , SO DEPRESSING A LACK

4 5 6 7 8 9 0 1 2 3 4

NUMBER OF CONFIGURATIONS FOUND : 2

```
**      PROTAN SYSTEM      UCL/PSP AT LOUVAIN-LA-NEUVE      PAGE:      4      **
**      PROGRAM: CWPAT      DATE: Mar11 93      TIME: 21:58:44      **
** HLM
```

<south (crass. lack)> /

TEXT WORD	SEQ NO	SEG NO
SOUTH	7	1
CRASS	26	1
CRASSNESS	63	1
LACK	94	2
SOUTH	215	4

SEG NO : 1
SEQ NO : 7

SOUTH TO-DAY , AND GEORGIA IS PERHAPS THE WORST .

7 8 9 0 1 2 3 4 5 6

THE ONE IS SIMPLY SENILE ; THE OTHER IS CRASS

7 8 9 0 1 2 3 4 5 6

NUMBER OF CONFIGURATIONS FOUND : 1

TEXT WORD	SEQ NO	SEG NO
SOUTH	7	1
NORTH	53	1
NORTH	68	2
NORTH	84	2
LACK	94	2
SOUTH	215	4

SEG NO : 2
SEQ NO : 84

NORTH IS THERE SUCH COMPLETE STERILITY , SO DEPRESSING A LACK

4 5 6 7 8 9 0 1 2 3 4

NUMBER OF CONFIGURATIONS FOUND : 1

```

**      PROTAN SYSTEM          UCL/PSP AT LOUVAIN-LA-NEUVE    PAGE:    5  **
**      PROGRAM: CWPAT         DATE: Mar11 93    TIME: 21:58:44  **
** HLM                                     **

```

```

(north south) lack./

```

TEXT WORD	SEQ NO	SEG NO
SOUTH	7	1
NORTH	53	1
NORTH	68	2
NORTH	84	2
LACK	94	2
SOUTH	215	4

```

SEG NO : 2
SEQ NO : 84

```

```

NORTH IS THERE SUCH COMPLETE STERILITY , SO DEPRESSING A LACK

```

```

4      5 6      7      8      9      0 1 2      3 4

```

```

NUMBER OF CONFIGURATIONS FOUND : 1

```

D. Préparation des données

Aucune.

E. Autres informations

Aucune.

F. Remarques

A noter que les mots-clés peuvent être aussi bien des mots simples que des marqueurs, #marquel# par exemple, ou tout autre item permis dans un fichier-source. Par exemple, #marquel# serait admis par la syntaxe de CWPAT; la barre oblique (/) serait aussi admise, de même que tous les signes de ponctuation. Cependant, si on utilise comme mots-clés les caractères de la syntaxe de CWPAT [(,), <, >, et /], il faut alors entourer ceux-ci d'une apostrophe de part et d'autre. (L'apostrophe doit même être redoublée lorsque l'apostrophe est elle-même le caractère recherché). On pourra donc avoir des syntaxes comme:

```

north ('/' ''') (crass.lack.) /   on cherche les passages où on
                                   trouve le mot north, les mots crass. ou lack.,
                                   et les signes / ou '.

```

G. Messages propres au programme

```

*CWPA821S  option "NARR=1" not provided to "CSCUT"
*CWPA822E  no keyword matched
*CWPA823S  syntax error in pattern definition
*CWPA824W  literal item will be truncated
*CWPA825S  maxtag exhausted for this pattern

```

XXIX. Le programme CWREFER

A. But du programme

Le fichier de type REFER crée les variables dont a besoin le programme CWSELECT. Il y a deux façons de générer un tel fichier REFER. Soit l'analyste repère de manière impressionniste les mots qu'il veut soumettre au programme CWSELECT, soit il recourt à une procédure plus standard, consistant par exemple à demander la liste de tous les mots répondant à un certain critère. C'est ce que fait CWREFER, qui donne la liste de tous les mots présentant une fréquence minimale déterminée par les paramètres du programme.

B. Exécution

1. Appel du programme

```
... CWREFER fn1 fn2
```

a. où fn1 est le nom d'un fichier de type CWREFER qui contient les paramètres et options pour les critères de sélection des mots. Ce fichier donnera son nom au fichier de type REFER qui contiendra les mots retenus pour l'analyse.

b. où fn2 est le filename du fichier de type "WORDS" créé par CSCUT, CRWSTRIP, CWEDIT ou CWADD. Ce fichier est le corpus à traiter, d'où sont extraits les mots contenus dans le fichier REFER que crée le programme CWREFER.

2. Entrées/sorties: Les entrées dans le programme CWREFER sont les entrées de base de tout programme dans PROTAN, soit un fichier de paramètres et options, qui d'ailleurs donne son nom au fichier "REFER" créé par le programme, et le texte lui-même, sous forme d'un fichier "WORDS".

La sortie, mis à part la liste imprimée, est surtout constituée d'un fichier "REFER" prêt à être traité par CWSELECT (sauf pour les mots à supprimer éventuellement).

3. Paramètres et options:

STAT = "STAT = 1" récapitule les définitions de segment demandées par l'utilisateur et donne les statistiques de tri, c'est-à-dire, par segment demandé, le nombre de mots différents et le nombre total de mots.
"STAT = 0" (le défaut) inhibe cette impression.

FMIN = Par défaut, 2 est la fréquence minimale pour qu'un mot du fichier "WORDS" soit sélectionné ("FMIN = 2", défaut). Ce seuil peut être abaissé ou augmenté selon les besoins ou hypothèses de l'utilisateur (alors, "FMIN = n", n pouvant être égal à 1, 2, 3, 4, ...)

FSEL = Par défaut, la fréquence minimale FMIN concerne la fréquence des mots par segment ("FSEL = 'S'", défaut). Si "FSEL = 'T'",

c'est alors la fréquence totale des mots
qui sert de seuil à FMIN.

PUNCH = "PUNCH = 1" perfore les mots retenus à raison d'un mot
par ligne selon le format de perforation des fichier
REFER, c'est-à-dire apostrophe, mot, apostrophe. Il
n'y a pas de création du fichier REFER si "PUNCH = 0".

C. Exemples

1. Exemple de procédure d'appel

a. "... CWREFER ORLEANS QUEBEC".

Le fichier "ORLEANS CWREFER" contient les paramètres et options de
ce programme; le texte analysé est, lui, rassemblé dans le fi-
chier-système "QUEBEC WORDS".

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "ORLEANS CWREFER" (entre <
et >);

```
<STAT = 1, FMIN = 20, FSEL = T, PUNCH = 1;>
```

On va sortir du fichier WORDS tous les mots qui ont une fréquence
de 20 ou plus dans le ou les textes. Il ne restera plus qu'à
éliminer les mots-fonctions qui n'auraient pas beaucoup de sens
dans une analyse factorielle.

D. Préparation des données

Une fois le fichier REFER constitué, il reste à le nettoyer
des mots tels que pronoms, mots-fonctions, et autres prépositions.

E. Autres informations

Aucune.

F. Remarques

Aucune.

G. Messages propres au programme

Aucun

XXX. Le programme CWSELECT

A. But du programme

Le but du programme CWSELECT est d'accomplir la stratégie non-catégorielle par excellence de l'analyse du contenu. Il faudrait dire, plus correctement, qu'avec ce programme, tout est mis en place pour permettre à l'utilisateur d'exécuter lui-même le type d'analyse (statistique) qu'il désire effectuer sur des données non-catégorielles. Concrètement, CWSELECT crée une matrice de "mots par segment" où les mots, en abscisse, sont pris comme variables, et les segments, en ordonnée, sont pris comme unités d'observation, chaque case de la matrice représentant la fréquence d'un mot dans un segment. Les "mots par segments" peuvent aussi être des "patterns de mots", et ces patterns sont formés selon la syntaxe du programme CWPAT. L'usage de tels "patterns de mots" dans un fichier REFER nécessite un fichier WORDS créé au moyen de l'option NARR=1.

Alternativement, CWSELECT crée une matrice de "segments par mot", c'est-à-dire exactement l'inverse de ce qui précède. À noter que beaucoup d'options du programme CWSELECT demandent que le fichier WORDS ait été créé dans le programme CSCUT avec l'option "NARR = 1"; ce sont notamment tous les fichiers de type SASSELSQ, SASSELNA, et leurs correspondants ASCII "PUNCHM" (ou EBCDIC pour le CMS), ainsi que l'option "PRINTNA = 1" ou "PRINTNA = 2".

B. Exécution

1. Appel du programme

```
... CWSELECT fn1 fn2 fn3
```

a. où fn1 est le filename d'un fichier de filetype "CWSELECT"; ce fichier contient les paramètres et les options du programme et donnera son nom aux fichiers de type "SASSEL" ou "SASSELOB" (et leur correspondant "PUNCHF" en ASCII ou EBCDIC), et "SASSELSQ" ou "SASSELNA" (et leur correspondant "PUNCHM" en ASCII ou EBCDIC). Ces derniers fichiers "SASSELSQ" et "SASSELNA" sont obtenus par l'option "PUNCH = 3" et "PUNCH = 4" respectivement; tous ces fichiers contiennent la matrice de "mots par segment" sous forme "perforée", prête à être traitée par l'un quelconque des programmes d'analyse statistique existant (ces programmes liront les fichiers PUNCHF ou PUNCHM) ou par le logiciel SAS (qui lira des fichiers de type SASSEL, SASSELOB, SASSELSQ, ou SASSELNA).

b. où fn2 est le filename du fichier de référence de filetype "REFER" contenant les différents mots que l'analyste souhaite voir inclus dans l'analyse statistique. Notons que le fichier REFER peut être constitué de:

- (1) de mots, bien évidemment, mais aussi
- (2) de marqueurs,
- (3) de signes de ponctuation,
- (4) de marqueurs non-verbaux tels qu'on

les utilise dans le programme CWADD,

(5) de séparateurs de n'importe quel type, comme '///' ou '--' par exemple (si l'analyste a attribué un sens quelconque à ces signes),

(6) et enfin de patterns de mots²⁴.

c. où fn3 est le filename du fichier de filetype "WORDS" créé par CSCUT, CRWSTRIP, CWEDIT ou CWADD. Ce fichier constitue le corpus à traiter, d'où sont extraits les mots repris dans le fichier de type "REFER" (fn2). Pour que les options "PRINTNA = 1" ou "PRINTNA = 2", et "PUNCH = 3" ou "PUNCH = 4" soient acceptées, ce fichier "WORDS" doit avoir été créé avec l'option "NARR = 1" au moment de l'exécution de CSCUT.

2. Entrées/sorties: Trois entrées sont nécessaires pour le fonctionnement du programme CWSELECT qui, par ailleurs, ne crée aucun fichier-système en sortie, mais seulement, et en option, des données perforées correspondant à la matrice des fréquences absolues et relatives (voir option PUNCH =).

La première entrée demandée est un fichier de filetype CWSELECT contenant les options. Ce même fichier donnera son nom au fichier de filetype "SASSEL" ou "SASSELOB" (ou "PUNCHF"), et "SASSELSQ" ou "SASSELNA" (ou "PUNCHM"), contenant la matrice des fréquences.

La deuxième entrée est le fichier de référence (de filetype REFER). Ce fichier contient les mots qui doivent entrer dans la matrice de fréquences. Ces mots (mais aussi marqueurs, signes de ponctuation, marqueurs non-verbaux, et tous types de séparateurs) doivent être présentés entourés d'une apostrophe de part et d'autre, et séparés par au moins un blanc (~) et/ou une virgule. Par défaut, 1.000 mots de référence sont autorisés, avec un maximum à 32.000. (Voir également la "Préparation des données", en particulier à propos de la syntaxe des patterns de mots).

La troisième entrée est le corpus lui-même, sous forme de fichier de type "WORDS" (issu de CSCUT --avec option "NARR = 1" si on utilise les options "PRINTNA = 1 (ou 2)" et "PUNCH = 3 (ou 4)"-- , CRWSTRIP, CWEDIT ou CWADD) d'où sont extraits les mots qui doivent entrer dans la matrice de fréquences.

3. Paramètres et options:

STAT = Si "STAT = 1", il y a impression des statistiques de tri; ces statistiques accompagnent les fichiers de type "WORDS". Il n'y a pas impression si "STAT = 0", qui est le défaut.

SORTW = Les mots du fichier de référence peuvent être triés en ordre alphabétique, ils ne doivent pas l'être. Si on souhaite qu'ils le soient, le paramètre "SORTW = 1" garantit un tri en ordre alphabétique de ces mots. "SORTW = 0" (le défaut) laissera les mots dans l'ordre où on les a mis dans le fichier de référence.

PRINT = A côté de "PRINT = 0", le défaut, qui n'imprime rien, le paramètre "PRINT = 1" permet l'impression de la matrice des fréquences absolues et relatives des mots.

²⁴ Tout comme dans le programme CWPAT, un "pattern de mots" dans CWSELECT peut contenir des mots, des marqueurs, des signes de ponctuation, des séparateurs, mais pas de marqueurs non-verbaux.

PUNCH = De même, à côté de "PUNCH = 0", le défaut, qui ne perfore rien, "PUNCH = 1" déclenche la perforation ("mots par segment") de la matrice des fréquences absolues et relatives sous forme d'un fichier de type "PUNCHF" ou "SASSEL". (Voir plus loin, "Autres informations" pour le format de lecture de ces fichiers). "PUNCH = 2" déclenche la perforation de la matrice transposée, c'est-à-dire des segments par mot. "PUNCH = 3" déclenche un type de perforation qu'on appelle "séquentiel": On trouve dans ce fichier, non pas le texte dans son ordre original, mais seulement les numéros de séquence des mots qui sont présents dans le fichier "REFER". Prenons l'exemple qui suit. Soit le texte:

"La présente note d'information s'adresse aux usagers de l'ordinateur IBM qui souhaitent utiliser le système de traitement différé VM Batch pour l'exécution de leurs programmes".

Si le fichier "REFER" avait été composé des mots suivants:

Fichier "REFER"	
'/1'	
'IBM'	
'note'	
'ordinateur'	
'pour'	
'/2'	
'utiliser'	
'programme'	
etc...	

le résultat de l'option "PUNCH = 3" aurait été:

Fichier "PUNCHM"					
...	nE de	nE de	nE de	nE de	nE de
segment	séquence	segment	séquence	segment	séquence
1	3 1	1	12 1	1	13 1
etc...					

Le numéro de séquence 3 correspond à la position du mot "note" qui est présent dans le fichier "REFER", de même que le mot "ordinateur" (position 12), et le mot "IBM" (position 13). On remarque que des groupes de mots de la liste "REFER" sont régulièrement précédés d'un code ('/1', puis '/2', etc... dans ce cas-ci), qui permet d'assigner chaque mot à un groupe d'items (à un facteur par exemple). Ce code peut aller de 1 à 999, il figure,

dans le PUNCH, à droite du numéro de séquence. Retenons encore que l'option "NARR = 1" doit avoir été fournie au programme CSCUT, tant pour "PUNCH = 3" que pour "PUNCH = 4".

L'option "PUNCH = 4" déclenche une perforation du texte dans son intégralité (perforation de type "narratif"), c'est-à-dire le niveau 0 (non-strippé) du fichier WORDS. Ici, chaque mot du texte est perforé, avec son indication de segment et de position (séquence dans le texte), et son code (0 si le mot est absent du fichier "REFER" et 999 si aucune indication n'est donnée). Dans les autres cas, le code correspond à ce qui figure en tête de chaque groupe d'items du fichier "REFER" (soit '/1', '/2', '/3', etc). Dans l'exemple ci-dessus, le résultat de "PUNCH = 4" aurait été:

Fichier "PUNCHM"									
...									
La	1	1	0	présente	1	2	0		
note	1	3	1	d	1	4	0		
information	1	5	0	s	1	6	0		
...				ordinateur	1	12	1		
IBM	1	13	1	...					

MAXREF = Par défaut, CWSELECT accepte de lire 1.000 mots de référence, le maximum étant de 32.000.

MAXPAT = concerne le nombre maximum de mots-clés qu'on peut mettre dans une recherche de patterns. Par défaut, 20 mots est le nombre de mots-clés acceptés dans chaque requête. Plus le nombre de mots-clés est élevé, moins il y a de chances de trouver des configurations contenant exactement ces mots-clés, et encore moins si on les demande dans un certain ordre.

MAXTAG = Par défaut, le nombre total de mots trouvés pour un pattern est de 1.000: c'est la valeur par défaut de MAXTAG. Lorsqu'on est en présence d'un corpus volumineux, on devra parfois accroître la valeur de MAXTAG à 5.000 ou 10.000 (peu de mémoire requise pour ce changement de paramètre). Par exemple, si on exploite le pattern

et: et /

dans un corpus de 100.000 mots, il est prudent de modifier la valeur de MAXTAG en conséquence, comme la fréquence d'un mot comme et atteint facilement 3.000 ou 4.000.

FIELD = Par défaut, on fait courir une "fenêtre" de recherche de patterns jusqu'à la limite du segment. Ce défaut s'écrit "FIELD = -1" (1 pouvant être aussi n'importe quel autre nombre négatif). Par contre, si *FIELD* est un nombre positif ("FIELD = 12" par exemple), l'étendue de la "fenêtre" de recherche est limitée à la valeur de "FIELD =".

EXTEND = Il est utile de préciser si la prochaine recherche de patterns (à l'intérieur d'une requête) doit commencer avec le dernier mot trouvé ("EXTEND = 0") ou bien juste après le dernier mot trouvé ("EXTEND = 1").

Toutes les options suivantes sont identiques à celles définies pour le programme CDWLOOK (soit HITLOW et HITUP correspondant à CATLOW et CATUP de CDWLOOK, SEGLOW et SEGUP, et SEQLOW et SEQUP) et ne doivent pas être redéfinies ici. Ces options sont reprises ci-dessous pour mémoire.

PRINTNA = Rappel: l'option "NARR = 1" dans CSCUT. Cfr. le programme CDWLOOK, page 10.7. A noter que si la comparaison entre le fichier REFER et le corpus se fait toujours sur la génération la plus récente du fichier WORDS, par contre l'impression via *PRINTNA* se fait en utilisant le niveau 0 du fichier WORDS, 'est-à-dire la sortie non-stripée de CSCUT.

LL =

HITLOW =

HITUP =

SEGLOW =

SEGUP =

SEQLOW =

SEQUP =

C. Exemples

1. Exemple de procédure d'appel

a. " ... CWSELECT DIX VINGT TRENTE".

Le fichier "DIX CWSELECT" contient les paramètres concernant la création de la matrice de fréquences des mots repris dans le fichier de référence "VINGT REFER". Ces mots font partie d'un corpus existant sous forme de fichier-système "TRENTE WORDS".

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "DIX CWSELECT" (entre < et >):

```
<
SORTW = 1, PRINT = 1, PUNCH = 1;
>
```

On demande que les mots du fichier de référence soient triés par le programme et que la matrice de fréquences des "mots par segment" soit imprimée et perforée.

3. Exemple de fichier de référence

a. Contenu du fichier "VINGT REFER" (entre < et >):

```
<
'FEMME' 'HEUREUX' 'FEU' 'VOULOIR'
>
```

D. Préparation des données

Les seules données à préparer sont les mots de référence qui doivent figurer, entourés d'une apostrophe de part et d'autre, dans le fichier de type REFER. Ce fichier peut être constitué, soit de façon manuelle, soit de façon automatique par le programme CWKWOC (avec l'option "PUNCH = 3"), soit de façon encore plus automatique par le programme CWREFER. On se reportera soit au point consacré au programme CWKWOC pour la procédure de constitution automatique d'un fichier de type REFER, soit au programme CWREFER.

Le contenu d'un fichier REFER est toujours reproduit au début du listing qui accompagne le programme CWSELECT. Un fichier REFER peut faire référence à des mots simples, comme 'ouvrier' ou 'chômage' par exemple, mais aussi à des patterns de mots, comme la séquence 'législation sociale'. Ces "patterns de mots" sont reproduits dans la première partie du fichier PUNCH juste après les commentaires.

Les patterns de mots obéissent à une syntaxe un peu particulière. Chaque pattern doit en effet être précédé d'un identificateur immédiatement suivi d'un double-point, par exemple:

lois-sociales: <(loi. législat.) soc.> / pattern de lois sociales

où

- ce qui est entre crochets **<(loi. législat.) soc.>** est le pattern dont le sens est "retrouver toutes les occurrences où les lemmes LOI ou LEGISLAT sont suivis du lemme SOC"; cette syntaxe va permettre de retrouver tous les passages où on trouve les séquences "loi(s) sociale(s)", "législation(s) sociale(s)", "loi(s) socialiste(s)", mais aussi "loi de protection sociale" si la fenêtre de recherche (le paramètre FIELD) est d'au moins 4 mots;
- "lois-sociales:" est l'identificateur, limité à 20 caractères au maximum, y compris le double-point;
- ce qui suit la barre oblique "/" est réservé à des commentaires éventuels.

Comme dans le programme CWPAT, l'opération logique "SUIVI DE" (c'est-à-dire des mots/racines inclus dans une paire de crochets) ne peut pas être incluse dans l'opération logique "OU" (c'est-à-

dire des mots/racines inclus dans une paire de parenthèses. Par contre, on peut additionner les résultats de patterns différents en les faisant précéder du même identificateur. Voici par exemple la séquence de quatre patterns suivants:

```
union:<union. douanière.> / commentaire: union suivi de douanière
union:<union. monétaire.> / commentaire: union suivi de monétaire
union: union-douanière. / commentaire: la racine "union-douanière"
union: union-monétaire. / commentaire: la racine "union-monétaire"
```

Si le premier pattern donne, disons, une fréquence de 5, le deuxième, une fréquence totale de 10, le troisième, de 6, et le quatrième, de 8, la fréquence totale (cumulée) sous l'identificateur **"union:"** sera de $5 + 10 + 6 + 8 = 27$.

E. Autres informations

L'option **"PRINT = 1"** permet l'impression de la matrice de fréquences des **"mots par segment"**. Sur la liste, on trouve, de gauche à droite, 1) l'indication du segment, suivie 2) du nombre total de mots dans le segment, 3) et du nombre de mots différents dans le segment, et enfin, 4) pour chaque mot, de ses fréquences absolue et relative dans le segment. Il est entendu que la fréquence relative d'un mot est le rapport, multiplié par 100, de la fréquence absolue de ce mot au nombre total de mots dans ce segment.

Les options **"PUNCH = 1"** et **"PUNCH = 2"** créent un fichier de type **"PUNCHF"** de **LRECL = 80**. Ce fichier est essentiellement destiné à entrer dans l'un ou l'autre programme d'analyse statistique. Voici la description de ce fichier de type **"PUNCHF"**:

PUNCHF
créé par CWSELECT
si PUNCH = 1 ou 2
(LRECL = 80)

1er enregistrement:

col 1 à 8	mot-clé "*PUNCHF~"
9 à 16	identification du fichier de sortie
17 à 24	date de création du fichier de sortie
25 à 32	heure de ...(idem)
33 à 40	identification du fichier WORDS en entrée
41 à 48	date de création du fichier WORDS en entrée
49 à 56	heure de ...(idem)
57 à 80	vierges

2me enregistrement:

col 1 à 5	mot-clé dépendant de la valeur attribuée à l'option PUNCH :
	"SEL~" : PUNCH = 1
	"SELOB" : PUNCH = 2
6 à 10	mot-clé "~POPT"
11 à 15	nombre d'options de traitement fournies à CSCUT
16 à 20	mot-clé "~GCOM"
21 à 25	nombre de lignes de commentaires généraux
26 à 30	mot-clé "~~OPT"
31 à 35	nombre d'options de traitement fournies au programme ayant généré ce fichier
36 à 40	mot-clé "~NSEG"
41 à 45	nombre de segments générés
46 à 50	mot-clé "~NREF"

51 à 55 nombre de mots de référence
 56 à 70 vierges
 71 à 75 mot-clé "~NPAT"
 76 à 80 nombre de lignes de requête de pattern
 3me enregistrement:
 col 1 à 64 le commentaire principal du fichier SOURCE
 (celui de l'enregistrement *SOURCE)
 65 à 80 vierges

les POPT enregistrements suivants:
 col 1 à 80 une option sous la forme XXX=YYY;

les GCOM enregistrements suivants:
 col 1 à 80 un commentaire général

les OPT enregistrements suivants:
 col 1 à 80 une option sous la forme XXX=YYY;

les NPAT enregistrements suivants:
 col 1 à 80 une ligne de requête de pattern

A. Pour PUNCH = 1 ("SEL~~")

enregistrements suivants (au nombre de (NREF+3)/4):
 col 1 à 80 par groupe de 20 colonnes, les NREF mots de
 référence

enregistrements suivants (NSEG enregistrements): chaque variable
 occupe 8 positions et est séparée de la suivante par un caractère de
 tabulation. Soit, pour chaque enregistrement:

- numéro du segment en cours (variable SEG)
- nombre total de mots dans le segment en cours

(variable WT)

- nombre de mots différents dans le segment en cours

(variable WD)

...l'ensemble suivant se répète autant de fois qu'il y a de
mots sélectionnés:

- fréquence absolue du premier mot pour le segment

en cours (variable F1)

- fréquence relative du premier mot pour le segment

en cours (variable RF1)

Si NREF>1, viennent alors les variables F2, RF2, etc.,

**... et passage à l'enregistrement suivant pour le 2ième
segment, etc.**

B. Pour PUNCH = 2 ("SELOB")

enregistrements suivants (NREF enregistrements): chaque variable
 occupe 8 positions (sauf le mot -WORD- qui en occupe 20) et est
 séparée de la suivante par un caractère de tabulation. Soit, pour
 chaque enregistrement:

- mot en cours (variable WORD)

**...l'ensemble suivant se répète autant de fois qu'il y a de
segments...**

- nombre total de mots dans le premier segment

(variable SWT1)

- nombre de mots différents dans le premier segment

(variable SWD1)

- fréquence absolue du mot en cours pour le premier
segment (variable SF1)

- fréquence relative du mot en cours pour le premier segment (variable SRF1)

si NSEG > 1), viennent alors les valeurs des variables SWT2, SWD2, SF2, SRF2, ...

... et passage à l'enregistrement suivant pour le 2ième mot, etc.

Note: La fréquence absolue est représentée par un nombre entier et la fréquence relative, par un nombre fractionnaire contenant trois décimales précédées du point décimal.

PUNCHM
créé par CWSELECT
si PUNCH = 3 ou 4
(LRECL = 80)

1er enregistrement:

col 1 à 8	mot-clé "*PUNCHM~"
9 à 16	identification du fichier de sortie
17 à 24	date de création du fichier de sortie
25 à 32	heure de ...(idem)
33 à 40	identification du fichier WORDS en entrée
41 à 48	date de création du fichier WORDS en entrée
49 à 56	heure de ...(idem)
57 à 80	vierges

2me enregistrement:

col 1 à 5	mot-clé dépendant de la valeur attribuée à l'option PUNCH: "SELSQ" : PUNCH = 3 "SELNA" : PUNCH = 4
6 à 10	mot-clé "~POPT"
11 à 15	nombre d'options de traitement fournies à CSCUT
16 à 20	mot-clé "~GCOM"
21 à 25	nombre de lignes de commentaires généraux
26 à 30	mot-clé "~~OPT"
31 à 35	nombre d'options de traitement fournies au programme ayant généré ce fichier
36 à 40	mot-clé "~NSEG"
41 à 45	nombre de segments générés
46 à 50	mot-clé "~NCAT"
51 à 55	nombre de catégories sélectionnées
56 à 60	mot-clé "~NTAG";
61 à 70	nombre de mots retenus + 1, pour PUNCH = 3 (SELSQ); nombre de mots retenus plus le nombre de mots non retenus, pour PUNCH = 4 (SELNA);
71 à 75	mot-clé "~NPAT"
76 à 80	nombre de lignes de requête de pattern

3me enregistrement:

col 1 à 64	le commentaire principal du fichier SOURCE (celui de l'enregistrement *SOURCE)
65 à 80	vierges

les POPT enregistrements suivants:

col 1 à 80	une option sous la forme XXX=YYY;
------------	-----------------------------------

les GCOM enregistrements suivants:

col 1 à 80	un commentaire général
------------	------------------------

les OPT enregistrements suivants:

col 1 à 80	une option sous la forme XXX=YYY;
------------	-----------------------------------

les NPAT enregistrements suivants:

col 1 à 80 une ligne de requête de pattern
enregistrements suivants (au nombre de (NCAT+19)/20):
col 1 à 80 par groupe de 4 colonnes, les NCAT codes de
groupe définis

A. Pour PUNCH = 3 ("SELSQ")

enregistrements suivants (NTAG enregistrements): ceux-ci sont rangées
en ordre croissant des numéros de séquence des mots et des numéros de
catégorie pour chaque numéro de séquence
Chaque variable occupe 8 positions et est séparée de la suivante par
un caractère de tabulation. Soit, pour chaque enregistrement:

- numéro du segment (variable SEG)
- numéro de séquence du mot retenu (variable
SEQUENCE)
- numéro de la catégorie (variable CAT)

**...et passage à l'enregistrement suivant pour la 2ième
observation, etc.**

Note: La dernière observation, fictive, est caractérisée par un
numéro de segment égal à NSEG+1, un numéro de séquence
égal au nombre total de mots +1 et un numéro de catégorie
nul.

B. Pour PUNCH = 4 ("SELNA")

enregistrements suivants (NTAG enregistrements): ils sont rangés en
ordre croissant des numéros de séquence des mots et des numéros de
catégorie pour chaque numéro de séquence. Chaque variable occupe 8
positions (sauf le mot -WORD- qui en occupe 20) et est séparée de la
suivante par un caractère de tabulation. Soit, pour chaque
enregistrement:

- un mot (variable WORD)
- numéro du segment (variable SEG)
- numéro de séquence du mot retenu (variable
SEQUENCE)
- numéro de la catégorie (variable CAT)

**... et passage à l'enregistrement suivant pour le 2ième
segment, etc.**

Cependant, l'option "SAS" de la procédure d'appel du pro-
gramme CWSELECT permet de convertir automatiquement ce fichier de
type "PUNCHF" en un fichier-système "SAS" de type "SASSEL" ou
"SASSELOB" (ou un fichier de type "PUNCHM" en fichier SAS de type
"SASSELSQ" ou "SASSELNA"). Quoi qu'il en soit, ce fichier est
destiné à être utilisé dans l'étape DATA d'un fichier SAS sous la
forme:

```
DATA;  
SET SASSEL.fn1;
```

où fn1 est le filename du fichier de paramètres de type CWSELECT.

Un tel fichier n'est visualisable que par la procédure PRINT
de SAS telle que:

```

CMS COPYFILE DIX SASSEL A = A (UNPACK;
DATA;
    SET SASSEL.DIX;
PROC PRINT LABEL;
PROC CONTENTS DATA=SASSEL.DIX;

```

pour le fichier perforé "DIX SASSEL" issu de la commande

```
" ... CWSELECT DIX VINGT TRENTÉ (PACK SAS)"
```

(cfr. exemple ci-dessus et remarquer la commande "CMS COPYFILE..." dans l'encadré, permettant de décondenser le fichier condensé).

Dans ces fichiers SAS, les variables de fréquence absolue associées aux n mots portent les noms F1, F2, ...F_n; les variables de fréquence relative associées aux n mots portent, elles, les noms RF1, RF2, ...RF_n. Les labels sont, eux, les mots précédés de F_ ou RF_, respectivement pour les fréquences absolues et relatives. Ce qui précède concerne le cas des fichiers de type "SASSEL". Dans le cas des fichiers de type "SASSELOB", les variables de fréquence absolue associées aux n segments portent les noms SF1, SF2, ...SF_n; les variables de fréquence relative portent les noms SRF1, SRF2, ...SRF_n. On trouve en outre une variable WORD, des variables SWT1, SWT2, ...SWT_n et SWD1, SWD2, ...SWD_n respectivement pour le nombre total de mots et le nombre de mots différents par segment. Les noms des autres variables sont détaillés dans la liste générée par la procédure CONTENTS de SAS.

F. Remarques

Comme suite au point précédent sur les fichiers de type "PUNCHF", "SASSEL", et "SASSELOB", voici le déroulement complet de la création d'un fichier de type "SASSEL" depuis la création du fichier "WORDS" jusqu'aux résultats de la procédure PRINT de SAS.

Une fois qu'a été exécuté le fichier "ROIS EXEC" contenant (les guillemets font partie du fichier²⁵ de type "EXEC"):

```

/* */
TRACE "C"
" ... CSCUT ROIS ROIS (NOSEND)"
IF RC > 8 THEN EXIT RC
" ... CRWSTRIP ROIS FRSTRP01 ROIS (NOSEND)"
IF RC > 8 THEN EXIT RC
" ... CRWSTRIP ROIS ROIS ROIS (NOSEND)"
IF RC > 8 THEN EXIT RC
" ... CWSELECT ROIS ROIS ROIS (SAS PACK)"
IF RC > 8 THEN EXIT RC
EXIT RC

```

et qui produit un fichier "ROIS SASSEL", on crée un fichier "ROIS SAS" contenant:

²⁵ Le fichier "ROIS REFER" de CWSELECT contient les mots de référence suivants <'\$E1GLISE' 'AVOIR' 'ET' 'E3TRE' 'DEVOIR-V' 'MAIS' 'PRINCE'>.

```

OPTIONS LINESIZE=68;
CMS COPYF ROIS SASSEL A = A (UNPACK;
DATA;
    SET SASSEL.ROIS;
PROC CORR PEARSON;
    VAR F1-F7 ;
PROC CONTENTS DATA=SASSEL.ROIS;

```

dont l'exécution donne le résultat suivant:

```

1          SAS(R) LOG  CMS SAS 5.16          VM/CMS USER BATCH7
NOTE: COPYRIGHT (C) 1984,1986 SAS INSTITUTE INC.,
          CARY, N.C. 27511, U.S.A.
NOTE: BATCH7 RELEASE 5.16 AT UNIVERSITE DE LOUVAIN (01089002).
NOTE: CPUID  VERSION = FF  SERIAL = 064075  MODEL = 4381 .
NOTE: SAS OPTIONS SPECIFIED ARE:
      DEFAULT=LOCAL LEAVE=64K SASNEWS=ANNONCE NONNEWS

```

```

1  OPTIONS LINESIZE=68;
2  CMS COPYF ROIS SASSEL A = A (UNPACK;
3  DATA;
4      SET SASSEL.ROIS;

```

NOTE: DATA SET WORK.DATA1 HAS 35 OBSERVATIONS AND 17 VARIABLES.
NOTE: THE DATA STATEMENT USED 0.15 SECONDS AND 216K.

```

5  PROC CORR PEARSON;
6      VAR F1-F7 ;

```

SAS				10:11 MONDAY, MARCH 28, 1988		
VARIABLE	N	MEAN	STD DEV	SUM	MINIMUM	MAXIMUM
F1	35	0.74286	1.40048	26.000	0	5.0000
F2	35	2.28571	2.26964	80.000	0	10.0000
F3	35	0.71429	1.29641	25.000	0	6.0000
F4	35	4.68571	2.85710	164.000	0	13.0000
F5	35	4.51429	3.21185	158.000	0	13.0000
F6	35	0.94286	1.02736	33.000	0	4.0000
F7	35	0.57143	1.17036	20.000	0	6.0000

SAS							10:11 MONDAY, MARCH 28, 1988	
PEARSON CORRELATION COEFFICIENTS								
/ PROB > !R! UNDER H0:RHO=0 / N = 35								
	F1	F2	F3	F4	F5	F6		
F1	1.00000	0.42168	-0.04166	0.08947	0.03026	0.05081		
F_ \$E1GLISE	0.0000	0.0116	0.8122	0.6093	0.8630	0.7719		
F2	0.42168	1.00000	0.15851	0.19114	0.20116	-0.00541		
F_AVOIR	0.0116	0.0000	0.3631	0.2714	0.2466	0.9754		
F3	-0.04166	0.15851	1.00000	0.24502	0.20585	0.42904		
F_DEVOIR-V	0.8122	0.3631	0.0000	0.1560	0.2355	0.0101		
F4	0.08947	0.19114	0.24502	1.00000	0.26172	0.38449		
F_ET	0.6093	0.2714	0.1560	0.0000	0.1288	0.0226		
F5	0.03026	0.20116	0.20585	0.26172	1.00000	0.48158		
F_E3TRE	0.8630	0.2466	0.2355	0.1288	0.0000	0.0034		
F6	0.05081	-0.00541	0.42904	0.38449	0.48158	1.00000		
F_MAIS	0.7719	0.9754	0.0101	0.0226	0.0034	0.0000		
F7	-0.06921	0.02531	0.55662	0.31916	0.17772	0.51718		
F_PRINCE	0.6928	0.8853	0.0005	0.0616	0.3071	0.0015		
F7								
F1	-0.06921							
F_ \$E1GLISE	0.6928							
F2	0.02531							
F_AVOIR	0.8853							
F3	0.55662							
F_DEVOIR-V	0.0005							
F4	0.31916							
F_ET	0.0616							
F5	0.17772							

F_E3TRE 0.3071
F6 0.51718
F_MAIS 0.0015
F7 1.00000
F_PRINCE 0.0000

NOTE: THE PROCEDURE CORR USED 0.25 SECONDS AND 280K
AND PRINTED PAGES 1 TO 2.

G. Messages propres au programme CWSELECT

*CWSE581S	duplicate refer item "xxxxxx"
*CWSE582W	refer item "xxxxxx" not matched
*CWSE583E	option "NARR = 1" not provided to "CSCUT" - options "PRINTNA" & "PUNCH" (3,4) set to zero
*CWSE584S	option "NARR=1" not provided to "CSCUT" - unable to process pattern references
*CWSE585W	no keyword matched
*CWSE586W	literal item will be truncated
*CWSE587S	syntax error in pattern definition
*CWSE588S	maxtag exhausted for this pattern

XXXI. Le programme CWTALLY

A. But du programme

CWTALLY a pour but de faire imprimer, avec leurs fréquences, tous les mots du corpus soit en ordre de fréquence inverse, c'est-à-dire des mots les plus fréquents aux mots les plus rares, soit en ordre de longueur inverse, c'est-à-dire des mots les plus longs aux mots les plus courts.

B. Exécution

1. Appel du programme

```
... CWTALLY fn1 fn2
```

a. où fn1 est le nom d'un fichier de filetype CWTALLY; ce fichier contient les paramètres et options, et, au minimum, le caractère syntaxique qui termine les séries d'options, ";".

b. et où fn2 est le nom du fichier-système de filetype "WORDS" créé par CSCUT, CRWSTRIP, CWEDIT ou CWADD.

2. Entrées/sorties: Aucun fichier-système n'est créé par ce programme qui appelle seulement, en entrée, outre le fichier de paramètres et options, un fichier-système de type "WORDS" créé par CSCUT, CRWSTRIP, CWEDIT, ou CWADD.

3. Paramètres et options:

STAT = Par défaut ("STAT = 0"), il n'y a pas impression des statistiques de tri, c'est-à-dire, par segment et pour le total, nombre de mots différents et nombre total de mots. "STAT = 1" déclenche l'impression de ces statistiques.

SORTT = Sans autre indication, CWTALLY imprime les mots selon un comptage sur l'ensemble des segments ("SORTT = 'T'"). L'option "SORTT = 'S'" permet une impression selon un comptage par segment.

TALLY = Par défaut ("TALLY = 'F'"), le tri porte sur la fréquence. "TALLY = 'L'" déclenche le tri sur la longueur des mots.

C. Exemples

1. Exemple de procédure d'appel

a. " ... CWTALLY PSYCHO NEURO".

Un fichier de paramètres "PSYCHO CWTALLY" et un fichier-système "NEURO WORDS" créé antérieurement par CSCUT ou CRWSTRIP sont tout ce qui est nécessaire à cette exécution.

2. Exemple de fichier de paramètres et options

a. Contenu du fichier "PSYCHO CWTALLY" (entre < et >):

```
<
STAT = 0, SORTT = 'T', TALLY = 'F';
>
```

Il y aura impression des mots triés selon la fréquence calculée sur le total des segments, mais les statistiques de tri ne sont pas demandées. On a ici toutes les options par défaut, et on aurait pu écrire seulement <;>.

D. Préparation des données

Aucune.

E. Autres informations

Une liste de mots en ordre de fréquence inverse n'a en soi pas grand intérêt, sinon de permettre d'établir divers indices de description lexicographique. Mais une telle liste présente un réel intérêt si on envisage d'utiliser l'approche non-catégorielle, et notamment le programme CWSELECT. Cette liste permet en effet de choisir les mots pleins les plus fréquents (en ignorant les mots-outils, en général encore plus fréquents) pour les entrer en abscisse comme variables dans une analyse multivariée en SAS, SPSS, ou autres logiciels de statistiques. Notons qu'on peut non seulement ignorer les mots-outils d'un corpus, mais les éliminer radicalement du fichier "WORDS" par CRWSTRIP.

L'intérêt d'une liste de mots rangés par ordre de longueur est d'illustrer d'éventuels indices lexicographiques de lisibilité (voir l'indice de lisibilité de Gunning dans le programme CSCUT).

F. Remarques

Aucune.

G. Messages propres au programme CWTALLY

Aucun

XXXII. Le programme CWWCOL

A. But du programme

CWWCOL est l'un des programmes d'analyse non-catégorielle du système PROTAN. CWWCOL se range donc au côté des programmes CWORD (d'une logique semblable mais plus complexe que CWWCOL) et CWSELECT.

La logique de CWWCOL est due à D. P. Spence (non publié). Le cadre théorique général du type de mesure qu'effectue CWWCOL est que les contraintes qui s'exercent sur le choix des mots dans un discours continu sont d'ordre sémantique et syntaxique sans doute, mais aussi d'ordre associatif et, dans cet ordre-là, relevant de processus largement inconscients: Le choix des mots dans une phrase est notamment de l'ordre de la fuite lexicale (Spence, Scarborough, & Ginsberg, 1978; Spence, 1979).

Lorsqu'un corpus peut être divisé en deux sous-ensembles --l'un dit "expérimental", et l'autre dit "de contrôle"-- selon une caractéristique quelconque, le but du programme CWWCOL est de comparer les deux sous-ensembles ainsi formés par rapport à un autre corpus, et d'identifier les mots qui différencient statistiquement les deux premiers sous-ensembles.

B. Exécution

1. Appel du programme

... CWWCOL fn1 fn2 fn3 [fn4...]

a. où fn1 est le nom d'un fichier (de filetype "CWWCOL") contenant les paramètres et les options.

b. où fn2 est le nom du fichier de référence. Le filetype de ce fichier doit être "WORDS".

c. où fn3, fn4, ... sont les noms des fichiers (de filetype WORDS) qui contiennent les parties du corpus qui vont être comparées entre eux. Il peut y avoir jusqu'à 9 fichiers de filetype "WORDS" différents.

2. Entrées/sorties: CWWCOL ne crée pas de fichiers--système en sortie, sinon des résultats sous forme de liste.

Mais en entrée, CWWCOL demande trois fichiers, dont le premier, "fn1 CWWCOL", ne nous occupera pas, non plus que le troisième (et suivants éventuels), "fn3 WORDS", qui contiennent le corpus à analyser.

C'est le deuxième fichier d'entrée "fn2 WORDS" qui demande quelque explication. Il s'agit donc de ce qu'on a appelé le "fichier de référence", c'est-à-dire le fichier par rapport auquel sont comparés les deux sous-ensembles du corpus. On l'a dit, ce fichier de référence doit se présenter sous forme d'un fichier de filetype "WORDS", mais ceci ne signifie pas qu'on doive référer à un texte se présentant sous forme d'un discours continu.

Mais d'abord, à quoi va-t-on, et que veut-on référer? Ici, deux possibilités se présentent. Pour commencer, on peut prendre comme fichier de référence la somme des deux sous-ensembles du corpus. En procédant de la sorte, on est assuré que tous les mots de chacun des deux sous-ensembles seront, sans exception, soumis à comparaison. Mais aussi, on peut, dans une autre optique, prendre comme fichier de référence, non plus la somme des deux sous-ensem-

bles, mais un autre corpus qui peut servir de point de référence valide. Un exemple ultérieur illustre ces deux stratégies.

Dans tous les cas, il faut savoir que les mots qui ont une fréquence inférieure à FMIN (voir point suivant), par segment ou au total (option FSEL, voir "Paramètres et options" au point suivant), dans le fichier de référence, ne sont pas pris en compte pour la comparaison. En outre, le système produit un message d'erreur si aucun mot du fichier de référence ne présente de fréquence supérieure ou égale à FMIN.

3. Paramètres et options: (dans le fichier "fn1 CWWCOL")

STAT = "STAT = 1" récapitule les définitions de segment demandées par l'utilisateur et donne les statistiques de tri, c'est-à-dire, par segment demandé, le nombre de mots différents et le nombre total de mots. "STAT = 0" (le défaut) inhibe cette impression.

FMIN = Par défaut, 2 est la fréquence minimale pour qu'un mot du fichier de référence soit pris en compte pour la comparaison ("FMIN = 2", défaut). Ce seuil peut être abaissé ou augmenté selon les besoins ou hypothèses de l'utilisateur (alors, "FMIN = n", *n* pouvant être égal à 1, 2, 3, 4, ...)

FSEL = Par défaut, la fréquence minimale FMIN de prise en compte des mots de référence concerne leur fréquence par segment ("FSEL = 'S'", défaut). Si "FSEL = 'T'", c'est alors la fréquence totale des mots de référence qui sert de seuil à FMIN.

PROBT = Par défaut ("PROBT = 0."), seules sont imprimées les valeurs *p* dont la probabilité est inférieure ou égale à 0.5. "PROBT = *n*", où *n* est plus grand que 0 et plus petit ou égal à 1, permet de modifier vers le haut ou vers le bas la limite en deçà de laquelle les résultats sont imprimés.

Puisque chaque segment doit appartenir soit au sous-ensemble expérimental, soit au sous-ensemble de contrôle, on va trouver ensuite, pour chaque segment, une définition de segment (en format libre), chaque définition devant tenir sur une seule ligne:

- a. identification du fichier "WORDS",
- b. numéro du segment (éventuellement en série, par exemple "1-4", voulant dire "segments numéros 1, 2, 3, et 4"),
- c. et l'appartenance du segment (éventuellement des segments) au groupe expérimental ou au groupe de contrôle (indexée par les lettres E ou

C respectivement),

d. avec un commentaire éventuellement.

On notera que les 4 éléments de définition de segment qui précèdent doivent, toujours sur la même ligne, être séparés d'au moins un blanc (~).

C. Exemples

1. Exemple de procédure d'appel

a. " ... CWWCOL DOUZE DOUZE DOUZE".

Un premier fichier "DOUZE CWWCOL", qui contient les paramètres, est suivi d'un deuxième "DOUZE WORDS" (le fichier de référence), qui est, dans le cas présent, exactement le même que le troisième fichier "DOUZE WORDS". Ceci signifie que le fichier de référence est en fait la somme des deux sous-ensembles et que, donc, tous les mots appartenant à chacun des deux sous-ensembles se trouveront dans le fichier de référence. On est ainsi assuré que la comparaison portera sur tous les mots du corpus (de fréquence supérieure à 1).

b. " ... CWWCOL DIAGNOS CANCER DIAGNOS".

Ici, le premier fichier "DIAGNOS CWWCOL" contient les paramètres du corpus sous forme du fichier "DIAGNOS WORDS", à analyser par le programme CWWCOL. En l'occurrence, il s'agit d'interviews de patients envoyés par différents médecins traitants en vue d'un diagnostic (par biopsie) de cancer cervical. Les interviews ont été effectuées par des personnes qui n'avaient pas connaissance de l'issue de la biopsie. Le contenu du fichier "DIAGNOS CWWCOL" révèle les interviews pour lesquelles, par ailleurs, un diagnostic positif a été prononcé (groupe expérimental), et les interviews pour lesquelles un diagnostic négatif (pas de cancer, groupe de contrôle) a été émis.

Ces deux sous-ensembles ont alors été comparés à un autre corpus, sous forme d'un fichier "CANCER WORDS", qui est en fait le journal intime d'une personne atteinte d'un cancer dans un stade terminal.

L'hypothèse est que le langage des interviewés qui ont un cancer cervical "sans le savoir" présentera plus de similitudes (c'est-à-dire moins de différences) avec le corpus de référence que le langage des interviewés pour lesquels le diagnostic est négatif.

2. Exemple de fichier de paramètres et options.

a. Contenu du fichier "DIAGNOS CWWCOL" (entre < et >):

```

<;
DIAGNOS 1      E  DIAGNOSTIC POSITIF
DIAGNOS 2      E  DIAGNOSTIC POSITIF
DIAGNOS 3      C  "    NEGATIF
DIAGNOS 4-8    E  "    POSITIF
DIAGNOS 9      C  "    NEGATIF
DIAGNOS 10     E  "    POSITIF
DIAGNOS 11-16 C  "    NEGATIF
>

```

Les interviews 1, 2, 4 à 8, et 10 appartiennent au groupe expérimental, et les interviews 3, 9, et 11 à 16, au groupe de contrôle. Un commentaire peut suivre les lettres E ou C.

D. Préparation des données

Il n'y a pas véritablement de préparation de données à faire, sinon à mettre le fichier de référence et le corpus à analyser, en forme de fichier de type "WORDS".

E. Autres informations

Ces informations concernent la présentation des résultats ainsi que la signification des mesures effectuées.

Dans l'ordre, on trouve

- 1) les options en vigueur,
- 2) les caractéristiques des fichiers de référence ("fn2 WORDS") et
- 3) du corpus ("fn3 WORDS", "fn4 WORDS", etc...),
- 4) les paramètres définis par l'utilisateur et enfin
- 5) les informations générées par l'option STAT.

Viennent ensuite les résultats proprement dits du programme CWWCOL. Ces résultats s'expriment essentiellement sous la forme de tests $\underline{t}$ avec leur probabilité correspondante. Le nombre de degrés de liberté pour ce test $\underline{t}$ figure d'ailleurs en tête de page "T TEST -- N.D.F. = $\underline{n}$ ".

On trouve ensuite différentes rubriques, dont voici d'abord l'énoncé, avant d'en donner le sens.

Ces rubriques sont, de gauche à droite sur la page:
 WORD / T STAT. / PROB / AVERAGE RATE OF WORD (EXP - CON) / SEG
 FREQ (EXP - CON) / WORD FREQ (EXP - CON) / .

WORD: Il s'agit d'un mot présent avec une fréquence supérieure ou égale à FMIN dans le fichier de référence et dans au moins un des segments du ou des fichiers WORDS à analyser. Les mots pour lesquels la valeur $\underline{t}$ est négative sont précédés de 3 astérisques (* * *).

WORD FREQ: La fréquence absolue du mot dans le groupe expérimental (EXP) et dans le groupe de contrôle (CON).

SEG FREQ: Il s'agit ici du nombre de segments, dans le groupe expérimental (EXP) et dans le groupe de contrôle (CON), où un mot est présent. Un mot peut avoir une fréquence de 20 par exemple dans le groupe expérimental, alors que cette fréquence de 20 est concentrée dans un seul segment; dans ce cas, "WORD FREQ = 20" et "SEG FREQ = 1". A l'opposé, un mot peut être présent une seule

fois dans chacun des 20 segments ("WORD FREQ = 20", "SEG FREQ = 20") et tout cela n'a pas la même signification.

AVERAGE RATE OF WORD: Le taux moyen de présence d'un mot dans le groupe expérimental (EXP) et de contrôle (CON). Chaque fréquence d'apparition d'un mot est transformée en une valeur:

$$tx = \text{SQRT} [(FREQMOT / NTOT) * 1000]$$

où FREQMOT = la fréquence du mot dans un segment;

où NTOT = le nombre total de mots dans le segment.

Le taux se calcule alors selon la formule:

Somme des tx / NSEG

où NSEG = le nombre de segments dans le groupe (EXP ou CON).

A titre d'exemple, dans le scénario "Douze hommes en colère" (cfr. fichier "DOUZE WORDS"), le mot "ACCUSATION" a un "AVERAGE RATE OF WORD (EXP)" de 0.474. L'"AVERAGE RATE OF WORD (CON)" est de 0.119. Ces valeurs sont obtenues de la façon suivante.

En sachant que les 9 segments du corpus ont été répartis en sorte que:

DOUZE	1-4	E	GROUPE EXPERIMENTAL
DOUZE	5-9	C	GROUPE DE CONTROLE

et que le mot "ACCUSATION" présente la distribution de fréquences suivante:

segment	fréquence	
1	2	
2	3	
3	0	soit WORD FREQ EXP = 5
4	0	
5	0	et
6	1	
7	0	WORD FREQ CON = 1
8	0	
9	0	

on a:

$0.474 = [\text{SQRT} ((2 / 1649) * 1000) + \text{SQRT} ((3 / 4755) * 1000)] / 4$ et $0.119 = [\text{SQRT} ((1 / 2824) * 1000)] / 5$

T STAT et *PROB*: sont, respectivement, la valeur du test t de Student et sa probabilité calculée en fonction du NDF (Number of degrees of freedom). Seules sont imprimées les valeurs de t dont la probabilité est inférieure ou égale à 0.50.

$$t = \frac{(Me - Mc)}{\sqrt{(Ne * SDe * SDe + Nc * SDc * SDc) * (Ne - Nc) / (NDF * Ne * Nc)}}$$

où Me = Moyenne du groupe expérimental = Se / Ne
 = AVERAGE RATE OF WORD (EXP)

où Se = Somme des valeurs tx du groupe expérimental
 Ne = Nombre de segments composant le groupe expérimental

(Même chose pour Mc, Sc, et Nc)

où SDe = Ecart-type du groupe expérimental
 = $\sqrt{(S2e / Ne) - Se * Se}$

où S2e = Somme des carrés des valeurs tx du groupe expérimental
 (Même chose pour SDc et S2c)

où NDF = Nombre de degrés de liberté du test $t = Ne + Nc - 2$

Le fait même du test t entraîne la conséquence qu'il faut au moins trois segments par analyse (--en d'autres mots, le nombre de degrés de liberté doit être strictement positif--), et qu'il faut au moins un segment dans chacun des groupes expérimental et de contrôle (--autrement dit, Ne et Nc doivent être strictement positifs--).

Cet ensemble de rubriques est se termine par un tableau récapitulatif, par niveau de probabilité, le nombre de valeurs t positives et négatives significatives.

F. Remarques

Aucune.

G. Messages propres au programme CWWCOL

```
*CWWC523S  segment definition missing
*CWWC524S  seg2 less than seg1
*CWWC525S  Exp/Con missing
*CWWC526S  Exp/Con syntax error
*CWWC527S  segment value > upper limit in file
*CWWC528S  identification length > 8
*CWWC529S  invalid char. in segment value
*CWWC532S  number of Exp/Con segments must be both > 0
*CWWC533S  total number of segments must be > 2
*CWWC534S  duplicate segment definition for
           file "xxxxxxx" -- segment nnnnn
*CWWC535S  no word selected in "REFER"
*CWWC536S  file identification not matched
*CWWC537S  duplicate file identification
*CWWC538W  file not referenced : "xxxxxxx (xxxxxxx)"
```

XXXIII. Le programme CWORD

A. But du programme

Comme CWWCOL, CWORD compare deux sous-ensembles d'un corpus, l'un dit expérimental, et l'autre dit "de contrôle", en termes d'un fichier tiers, dit "de référence". Mais ici, et à la différence de ce qui se passe pour CWWCOL, le fichier de référence est composé de mots choisis par l'utilisateur en fonction de ses hypothèses.

Mais CWORD va encore plus loin que CWWCOL dans le sens de la comparaison. Il est en effet possible de sous-diviser les deux sous-ensembles du corpus en plusieurs sous-groupes, et cela jusqu'à neuf sous-groupes différents.

Enfin, comme pour CWWCOL, le cadre théorique du programme CWORD a été défini par Spence (Spence, Scarborough, & Ginsberg, 1978; Spence, 1979): Le choix des mots dans une phrase relève de processus largement inconscients et est de l'ordre de la fuite lexicale.

B. Exécution

1. Appel du programme

```
... CWORD fn1 fn2 fn3 [fn4...]
```

a. où fn1 est le nom d'un fichier de filetype "CWORD" contenant paramètres et options,

b. où fn2 est le nom du fichier de référence, qui doit être de filetype "REFER",

c. et où fn3, fn4, etc... sont les noms des fichiers (de filetype "WORDS") contenant les textes qui vont être comparés les uns aux autres. Il peut y avoir jusqu'à neuf fichiers de type "WORDS" différents. Les identifications de ces fichiers doivent être définies dans le fichier "fn1 CWORD".

2. Entrées/sorties: CWORD ne crée pas de fichier--système en sortie, sinon des résultats sous forme de listes.

En entrée, CWORD requiert trois fichiers différents, soit "fn1 CWORD" qui contient les paramètres et les options, "fn2 REFER", qui est le fichier de référence contenant les mots qui vont servir à faire la comparaison, et "fn3 WORDS", qui contient les segments qui vont être soumis à la comparaison; le corpus peut être réparti dans plusieurs fichiers "fn4 WORDS", "fn5 WORDS", etc.

On retiendra encore que les différents mots de référence doivent être mis entre apostrophes, et être séparés par au moins un blanc (~) et/ou une virgule.

Le choix des mots qu'on va mettre dans le fichier de référence "fn2 REFER" est question d'appréciation. En réalité, le groupement de mots qu'on met dans le fichier de référence doit avoir un rapport avec le découpage du corpus. Autrement dit, il faut que chacun des mots et l'ensemble des mots du fichier de référence créent le plus de contraste possible entre les deux parties du corpus ainsi découpé.

3. Paramètres et options: (dans le fichier "fn1 CWORD")

`STAT =` "STAT = 1" récapitule les définitions de segment demandées par l'utilisateur et donne les statistiques de tri, c'est-à-dire, par segment demandé, le nombre de mots différents et le nombre total de mots. "STAT = 0" (le défaut) inhibe cette impression.

Le fichier "fn1 CWORD" comporte ensuite les paramètres qui gèrent le découpage du corpus. On va trouver pour chaque segment une définition de segment (en format libre), chaque définition devant tenir sur une seule ligne:

- a. identification d'un fichier de type "WORDS",
- b. un numéro de segment, éventuellement des numéros de segments en série, par exemple "1-4", voulant dire "segments 1 à 4",
- c. l'appartenance du segment (ou des segments) au groupe expérimental (lettre E) ou au groupe de contrôle (lettre C),
- d. l'appartenance de chaque segment, ou série de segments, à un sous-groupe éventuel, indexée par un chiffre allant de 1 à 9; si aucun sous-groupe n'est prévu ou possible, le chiffre 0 désigne l'appartenance à l'échantillon total,
- e. un commentaire éventuellement.

Les éléments de définition de segment doivent être séparés par au moins un blanc (~). Si une division en sous-groupes est prévue, il est alors possible d'assigner à chaque sous-groupe ainsi constitué une étiquette qui accompagnera chacune des opérations statistiques ultérieures. Cette définition d'étiquette est générée par les mots-clés suivants:

- (1) un caractère "/" (barre oblique),
- (2) un chiffre de 0 à 9, représentant soit l'échantillon total (0), soit l'un des neuf sous-groupes possibles,
- (3) l'étiquette elle-même, d'ailleurs optionnelle, de 30 caractères,
- (4) et un commentaire, également optionnel, se plaçant après le trentième caractère de l'étiquette.

Ici aussi, chacun des éléments de la définition d'étiquette doit figurer sur une même ligne; le chiffre et l'étiquette doivent être séparés par au moins un blanc (~). Notons que les définitions d'étiquette peuvent être placées n'importe où dans le fichier.

C. Exemples

1. Exemples de procédure d'appel

" ... CWORD ELECTION ELECTION ELECTION".

Le premier fichier "ELECTION CWORD" (fn1 avec les définitions de segments) est suivi du deuxième fichier de référence "ELECTION REFER" (fn2 avec les mots de la comparaison), et du troisième fichier "ELECTION WORDS" (fn3 avec le corpus à analyser).

2. Exemple de fichier de référence

a. Contenu du fichier "ELECTION REFER" (entre < et >):

```
<
'AVENIR' 'CLAIR' 'CONFIANCE' 'RESULTAT'
>
```

Les quatre mots ci-dessus sont retenus pour la comparaison des deux parties du corpus contenues dans le fichier "ELECTION WORDS". Les définitions de segments figurent dans le fichier "ELECTION CWORD".

3. Exemple de fichier de paramètres (définitions de segments)

a. Contenu du fichier "ELECTION CWORD" (entre < et >):

```
<;
ELECTION 1 E 2 PVV LIBERAL FLAMAND
ELECTION 2 E 1 PSC SOCIAL CHRETIEN FRANCOPHONE
ELECTION 3 E 2 CVP SOCIAL CHRETIEN FLAMAND
ELECTION 4 C 2 VB VLAAMS BLOK
ELECTION 5 C 1 ECOLO ECOLOGISTE FRANCOPHONE
ELECTION 6 C 2 VU VOLKSUNIE
ELECTION 7 C 2 SP SOCIALISTE FLAMAND
ELECTION 8 C 2 AGALEV ECOLOGISTE FLAMAND
ELECTION 9 C 1 PC COMMUNISTE
ELECTION 10 E 1 PRL LIBERAL FRANCOPHONE
ELECTION 11 C 1 UDRT RADICAL
ELECTION 12 C 1 FDF FRONT DES FRANCOPHONES
ELECTION 13 C 1 PS SOCIALISTE FRANCOPHONE
/ 0 ENSEMBLE DES PARTIS
/ 1 PARTIS FRANCOPHONES PSC ECOLO PC PRL UDRT FDF PS
/ 2 PARTIS FLAMANDS PVV CVP VB VU SP AGALEV
>
```

Lors des élections de septembre 1985 en Belgique, le journal "Le Soir" a laissé 13 présidents de parti politique présenter leurs vues politiques dans une chronique hebdomadaire intitulée "Carte blanche". On a rassemblé les 13 textes en les répartissant de l'interview 001 à l'interview 013. Ceci a donné lieu à 13 segments auxquels a été attribué le code E (pour Expérimental) lorsque le segment correspond à un parti de la majorité, ou le code C (pour Contrôle) lorsque le segment correspond à un parti de l'opposition.

Mais on a aussi assigné un code 1 ou 2 (c'est-à-dire sous-groupe 1 ou 2) selon que le segment correspond à un parti francophone ou à un parti flamand. Ce qui suit la quatrième colonne est à considérer comme commentaire.

Afin d'identifier aisément les sous-groupes, on a aussi assigné une étiquette "PARTI FRANCOPHONE" ou "PARTI FLAMAND" aux sous-groupes 1 et 2, à nouveau avec des commentaires à partir du trentième caractère après le début de la zone d'étiquette.

D. Préparation des données

Il n'y a pas vraiment de données à préparer, sinon à disposer du corpus sous forme de fichier de type "WORDS". Par contre, la constitution du fichier de référence peut demander un certain travail de réflexion et d'analyse.

Par exemple, les quatre mots du fichier "ELECTION REFER" ont été sélectionnés après l'exécution du programme CWWCOL sur le même fichier "ELECTION WORDS". Cette analyse par CWWCOL a révélé en effet que ces quatre mots étaient de ceux qui, tout en présentant en général des fréquences non nulles de part et d'autre, différenciaient statistiquement les deux séries de textes. On aura l'occasion plus loin d'examiner les résultats obtenus à l'aide d'un autre fichier de référence, composé des trois mots "NOS", "NOUS", et "NOTRE".

Notons que le programme CWKWOC (avec l'option "PUNCH = 3") produit automatiquement un fichier de type "REFER" immédiatement utilisable par le programme CWORD.

E. Autres informations

Les résultats se présentent en partie de manière semblable à ceux du programme CWWCOL.

On trouve donc, dans l'ordre,

- a. les options en vigueur, suivies
- b. des caractéristiques du ou des fichiers "fn3 WORDS", "fn4 WORDS", etc.,
- c. les paramètres demandés par l'utilisateur,
- d. le tableau associé à l'option STAT, et enfin,
- e. les résultats proprement dits du programme CWORD exprimés sous forme de tests t avec les nombres de degrés de liberté (N.D.F.) et la probabilité correspondante.

1. Tests t sur les mots de référence

Les rubriques qui détaillent les résultats sont, de gauche à droite sur la page: WORD / SUBGROUP NUMBER & IDENTIFICATION / T STAT / N.D.F. / PROB. / AVERAGE RATE OF WORD (EXP - CON) / SEG FREQ (EXP - CON) / WORD FREQ (EXP - CON) / :

WORD: On trouve ici chacun des mots qui figurent dans le fichier de référence.

SUBGROUP NUMBER & IDENTIFICATION: Chacun des mots de référence est répété autant de fois qu'il y a de sous-groupes identifiés (par un code de 1 à 9); même si aucun sous-groupe n'a été généré, le code "0" identifie toujours l'échantillon total. Si aucune étiquette n'a été écrite pour le code "0", ce dernier est suivi par défaut de l'étiquette "TOTAL SAMPLE".

WORD FREQ: La fréquence absolue du mot de référence dans le groupe expérimental (EXP) et dans le groupe de contrôle (CON), à l'intérieur de l'échantillon total et de chacun des sous-groupes identifiés. Dans le cas d'une fréquence proche de zéro, ou égale à zéro, dans l'un ou l'autre groupe, il y aurait peut-être lieu de se montrer prudent dans l'interprétation d'un test t significatif.

SEG FREQ: Le nombre de segments, dans le groupe expérimental (EXP) et dans le groupe de contrôle (CON), et par sous-groupe, où un mot de référence est présent. Ici aussi, dans le cas d'une fréquence proche de zéro, ou égale à zéro, dans l'un ou l'autre

groupe, il y aurait lieu de se montrer prudent dans l'interprétation d'un test t significatif.

AVERAGE RATE OF WORD: Le taux moyen de présence d'un mot de référence dans le groupe expérimental (EXP) et dans le groupe de contrôle (CON). Chaque fréquence est transformée en une valeur:

$$t_x = \text{SQRT} [(FREQMOT / NTOT) * 1000]$$

où FREQMOT = la fréquence du mot dans un segment;
où NTOT = le nombre total de mots dans le segment.

Le taux se calcule alors selon la formule:

Somme des t_x / NSEG

où NSEG = le nombre de segments dans le groupe (EXP ou CON).

A titre d'exemple, dans le corpus "ELECTION", le mot de référence "NOUS" --tiré du fichier de référence composé de "NOUS", "NOS", et "NOTRE"-- présente les valeurs suivantes:

			AVERAGE RATE OF WORD	
			EXP	CON
NOUS	0	TOTAL SAMPLE	3.626	2.116
NOUS	1	PARTI FRANCOPHONE	2.916	2.324
NOUS	2	PARTI FLAMAND	4.337	1.856

Ces valeurs ont été obtenues de la façon suivante. On sait que le corpus a été divisé en 4 segments expérimentaux (E) et 9 segments de contrôle (C). Connaissant la valeur NTOT (le nombre total de mots par segment) et la valeur FREQMOT (la fréquence du mot de référence, "NOUS" dans ce cas-ci) par segment²⁶, on a le tableau suivant:

Mot de référence "NOUS"					
SEGMENT	SERIE	SOUS-GROUPE	ETIQUETTE	FREQMOT	NTOT
1	E	2	PVV	8	521
2	E	1	PSC	5	907
3	E	2	CVP	18	796
4	C	2	VB	2	800
5	C	1	ECOLO	3	639
6	C	2	VU	3	725
7	C	2	SP	5	634
8	C	2	AGALEV	1	999
9	C	1	PC	2	699
10	E	1	PRL	5	412
11	C	1	UDRT	6	497
12	C	1	FDF	1	848
13	C	1	PS	7	683
				66	

Le calcul de la valeur AVERAGE RATE OF WORD du mot "NOUS" dans le groupe expérimental pour l'échantillon total se déroule comme suit:

²⁶ Ces renseignements sont donnés par CSCUT ou CRWSTRIP.

$$3.626 = [\text{SQRT}((8/521)*1000) + \text{SQRT}((5/907)*1000) + \text{SQRT}((18/796)*1000) + \text{SQRT}((5/412)*1000)] / 4 \\ = [3.919 + 2.348 + 4.755 + 3.484] / 4 = 14.506 / 4$$

Et le calcul de la valeur AVERAGE RATE OF WORD du mot "NOUS" dans le groupe expérimental pour le sous-groupe "PARTI FLAMAND" se déroule de la même façon, soit:

$$4.337 = [\text{SQRT}((8/521)*1000) + \text{SQRT}((18/796)*1000)] / 2 \\ = [3.919 + 4.755] / 2$$

T STAT, *N.D.F.*, et *PROB*: sont, respectivement, la valeur du test $\underline{t}$ de Student, le nombre de degrés de liberté, et la probabilité de la valeur $\underline{t}$, pour l'échantillon total et chaque sous-groupe de chaque série, expérimentale et de contrôle²⁷.

$$t = \frac{(Me - Mc)}{\text{SQRT}((Ne * SDe * SDe + Nc * SDc * SDc) * (Ne - Nc) / (NDF * Ne * Nc))}$$

où Me = Moyenne du groupe expérimental = Se / Ne
= AVERAGE RATE OF WORD (EXP)

où Se = Somme des valeurs $\underline{tx}$ du groupe expérimental
Ne = Nombre de segments composant le groupe expérimental
(Même chose pour Mc, Sc, et Nc)

où SDe = Ecart-type du groupe expérimental
= SQRT((S2e / Ne) - Se*Se)

où S2e = Somme des carrés des valeurs $\underline{tx}$ du groupe expérimental
(Même chose pour SDc et S2c)

où NDF = Nombre de degrés de liberté du test $\underline{t}$ = Ne + Nc - 2

Le fait même du test $\underline{t}$ entraîne la conséquence qu'il faut au moins trois segments par analyse --en d'autres mots, le nombre de degrés de liberté doit être strictement positif-- et qu'il faut au moins un segment dans chacun des groupes expérimental et de contrôle, ces contraintes existant en outre pour chaque sous-groupe.

²⁷ Dans l'exemple cité du fichier ELECTION, ceci permet de constater que, pour l'échantillon total, la valeur $\underline{t}$ du mot de référence "NOUS" est de 2.737, ndf = 11, $\underline{p} < .05$, mais qu'en réalité, cette différence significative est due uniquement aux partis flamands ($\underline{t}$ = 3.957, ndf = 4, $\underline{p} < .05$), la valeur $\underline{t}$ pour les partis francophones étant de $\underline{t}$ = 0.730, ndf = 5, $\underline{p} < .50$. Ce sont les partis flamands de la majorité (PVV et CVP) qui tendent à utiliser plus souvent la référence à la "nostrité" si on veut. On pourrait ajouter, à titre de commentaire, que cet usage de la nostrité est une source de sophismes ouvrant à beaucoup de malentendus dans la mesure où cet usage permet d'évoquer, sans précisions, le sens distributif, ou le sens idéal, voire moyen, de la nostrité, ou encore son sens collectif indivisible.

Notons encore que la valeur de ce test $\underline{t}$ sur les mots de référence est remplacée par des étoiles lorsque les écarts-types des deux séries expérimentale et contrôle sont nuls (et a fortiori lorsque la fréquence d'un mot de référence est nulle de part et d'autre, le mot n'ayant pas été trouvé).

Suit enfin un tableau récapitulatif des valeurs de probabilité des tests $\underline{t}$, positives et négatives, pour l'échantillon total et chaque sous-groupe.

2. Tests $\underline{t}$ sur taux, dispersion, et total

Les tests sur les mots de référence pris isolément peuvent être répétés sur l'ensemble des mots de référence considéré comme construction homogène. De même que le point de départ des tests sur les mots de référence isolés est leur distribution dans les segments, de même, le point de départ des tests sur le groupe de mots de référence est la distribution de ce groupe dans les segments.

De gauche à droite de la page, on trouve alors les rubriques suivantes:

/ FILE ID. / SEG / EXP - CON / SUBGROUP NUMBER & IDENTIFICATION / RATE / RANGE / TOTAL /.

FILE ID.: soit l'identification du fichier "WORDS".

SEG: soit le numéro du segment.

EXP - CON: soit l'appartenance de chaque segment à la série expérimentale ou de contrôle.

SUBGROUP NUMBER & IDENTIFICATION: soit l'appartenance de chaque segment à un sous-groupe éventuel, sinon à l'échantillon total.

TOTAL: par segment, la somme des fréquences de tous les mots de référence considérés.

RANGE: l'indice de dispersion est le nombre de mots de référence différents trouvés par segment.

RATE: le taux est égal à $[\text{SQRT} ((\text{TOTAL} / \text{NTOT}) * 1000)]$ où NTOT est encore une fois le nombre total de mots dans le segment.

A partir de ces statistiques descriptives d'usage du groupe de mots de référence, on peut alors à nouveau effectuer les mêmes tests $\underline{t}$ sur le groupe de mots de référence que ceux qu'on a fait précédemment pour chaque mot de référence pris isolément. Et avec des rubriques identiques ou comparables:

BASE: prend la place du mot de référence pris isolément et couvre les trois indices d'usage calculés plus haut, soit RATE, RANGE, et TOTAL.

SUBGROUP NUMBER & IDENTIFICATION: comme précédemment, autant qu'il y a de sous-groupes considérés.

WORD FREQ: la somme des fréquences des mots de référence, par série et par sous-groupes.

SEG FREQ: par série et par sous-groupe, la somme des segments où au moins un mot de référence a été trouvé.

AVERAGE RATE OF BASE: par série et par sous-groupe, la moyenne des indices RATE, RANGE, et TOTAL.

T STAT, N.D.F., PROB: comme précédemment, mais sur les indices RATE, RANGE, et TOTAL.

En poursuivant l'exemple commencé plus haut sur le fichier "ELECTION", nous avons les statistiques d'usage suivantes (rappel: le fichier de référence est composé de NOUS + NOS + NOTRE):

SEGMENT IDENT	SUBGROUP IDENT	RATE	RANGE	TOTAL
ELECTION 1	2 PARTI FLAMAND	4.595	3	11
ELECTION 2	1 PARTI FRANCOPHONE	2.778	2	7
ELECTION 3	2 PARTI FLAMAND	5.931	2	28
ELECTION 4	2 PARTI FLAMAND	1.936	2	3
ELECTION 5	1 PARTI FRANCOPHONE	2.797	2	5
ELECTION 6	2 PARTI FLAMAND	2.034	1	3
ELECTION 7	2 PARTI FLAMAND	3.552	3	8
ELECTION 8	2 PARTI FLAMAND	1.733	3	3
ELECTION 9	1 PARTI FRANCOPHONE	1.692	1	2
ELECTION 10	1 PARTI FRANCOPHONE	4.674	3	9
ELECTION 11	1 PARTI FRANCOPHONE	4.255	3	9
ELECTION 12	1 PARTI FRANCOPHONE	1.881	3	3
ELECTION 13	1 PARTI FRANCOPHONE	3.630	3	9

Par exemple, dans le segment 1, la valeur RATE de 4.595 est obtenue selon la formule:

$$\text{SQRT} [(11 / 521) * 1000] = 4.595$$

(1) où 11 est la somme des fréquences de NOUS + NOS + NOTRE dans le segment 1

(2) et où 521 est le nombre total de mots dans le segment 1.

Selon un scénario maintenant connu, le calcul de la valeur AVERAGE RATE OF BASE pour la base RATE dans l'échantillon total (groupe expérimental) se déroule comme suit:

$$(4.595 + 2.778 + 5.931 + 4.674) / 4 = 4.494$$

De même, la valeur AVERAGE RATE OF BASE pour la base RANGE du sous-groupe 1 (PARTI FRANCOPHONE) (groupe expérimental) est de

$$[(2 + 3) / 2 = 2.5].$$

On constate que pour l'échantillon total, la valeur $\underline{t}$ de l'ensemble des mots de référence NOUS + NOS + NOTRE est de

$$\underline{t} = 2.920 \text{ (Ndf} = 11, \underline{p} < .05)$$

pour la base RATE; que pour les partis flamands, cette valeur est de $\underline{t} = 3.942$ (ndf = 4, $\underline{p} < .05$) pour la même base RATE, et que pour les partis francophones, $\underline{t} = 0.906$ (ndf = 5, $\underline{p} < .50$) toujours pour la même base.

Ce qu'on a constaté plus haut concernant l'usage du mot NOUS par les partis flamands de la majorité se confirme ici en s'étendant à la "nostrité", c'est-à-dire les mots NOUS, NOS, et NOTRE. Un résultat similaire s'observe pour la base TOTAL, mais pas pour la base RANGE: Ceci est compréhensible vu la variabilité limitée du groupe des mots de référence constitué de seulement trois mots.

F. Remarques

Aucune.

G. Messages propres au programme CWORD

*CWWO551S	subgroup number missing
*CWWO552S	subgroup number syntax error
*CWWO553S	segment definition missing
*CWWO554S	seg2 less than seg1
*CWWO555S	Exp/Con missing
*CWWO556S	Exp/Con syntax error
*CWWO557S	segment value > upper limit in file
*CWWO558S	identification length > 8
*CWWO559S	invalid char. in segment value
*CWWO562S	number of Exp/Con segments must be both > 0
*CWWO563S	total number of segments must be > 2
*CWWO564S	duplicate segment definition for file "xxxxxxx" -- segment nnnnn
*CWWO566S	file identification not matched
*CWWO567S	duplicate file identification
*CWWO568W	file not referenced : "xxxxxxx (xxxxxxx)"

XXXIV. Annexes

A. Les dictionnaires de formes nominales et de catégories

On ne dispose à ce jour²⁸, pour la langue française, que d'un seul dictionnaire de formes nominales de 8.908 entrées, élaboré par Hogenraad (voir Hogenraad & Orianne, 1983). On y réfèrera dorénavant sous le nom de "liste FRSTRP01" (FR pour "FRench"); cette liste est appelée sous le nom de FRSTRP01. Il s'agit d'un ensemble de mots d'usage courant (verbes conjugués, adjectifs et adverbes, noms) qui, tous, sont ramenés à leur forme nominale (ou minimale), c'est-à-dire infinitif pour les verbes, masculin singulier pour les adjectifs, et singulier pour les noms, et parfois à leur radical. L'exécution du programme CRWSTRIP au moyen d'une liste comme la liste FRSTRP01 doit normalement réduire quelque peu le nombre de mots différents dans un texte courant. Les mots qui ne seraient pas modifiés par cette liste peuvent bien sûr toujours l'être soit par addition de nouvelles entrées à la liste standard existante, soit par création d'une liste correspondant aux besoins spécifiques de chaque utilisateur; le programme CWKWOC peut ici être utilisé pour ajouter de nouvelles entrées au fichier existant ou pour créer un fichier séparé. Il est en effet toujours possible de faire passer plusieurs fois un même texte --en réalité, il faut dire un même fichier de type "WORDS"-- par le programme CRWSTRIP en utilisant chaque fois un dictionnaire de formes nominales différent, et cela jusqu'à ce que l'utilisateur s'estime satisfait du nombre de mots différents dans son corpus. Toujours pour la langue française, on dispose également d'un dictionnaire de formes nominales de 175 entrées (nom = "FRSTRP11") composé uniquement de mots-outils à supprimer du corpus.

On dispose aussi d'un dictionnaire de formes nominales de 5.469 entrées pour la langue allemande, élaboré par Arnold Guntern (1981). C'est une liste que les programmes appellent sous le nom de GGSTRP01 (GG pour German Germany). La liste de langue allemande est appelée "liste GGSTRP01" dans ce manuel.

On dispose également d'un dictionnaire de formes nominales de 6.245 entrées pour la langue anglaise; cette liste, qui faisait partie du système WORDS de H. P. Iker, a été adaptée au présent système PROTAN. Les programmes appellent cette liste par le nom de AESTRP01 (AE pour American English), et c'est sous le nom de "liste AESTRP01" qu'on y réfèrera ici. Un autre dictionnaire, de 141 mots-outils de l'anglais, porte le nom "AESTRP11".

On dispose enfin de deux dictionnaires réduits de formes nominales, l'un, "ITSTRP01" de 232 entrées en italien; l'autre, "SPSTRP01", de 1.103 entrées pour l'espagnol, d'ailleurs accompagné d'un autre dictionnaire "SPSTRP11" de 21 entrées, composé de mots-outils espagnols à supprimer.

Par contre, on dispose, pour le programme CDWLOOK, de dictionnaires de catégories variés et nombreux. On trouvera ici un aperçu du contenu et de l'origine de chacun des dictionnaires disponibles, à commencer par ceux en langue française:

1. Le dictionnaire **DIMAB** est un dictionnaire basé sur des normes d'imagerie. Le DIMAB a été élaboré par Hogenraad et

²⁸ Tous les dictionnaires présentés ici sont de type "ROOTS", "DICTS" ou "FINDS" et ne doivent plus être traités par les programmes CRCHECK, CDCHECK, ou CFCHECK.

Orianne (1981). On y trouve les normes d'imagerie, étalées de la catégorie 1 (imagerie faible) à la catégorie 7 (imagerie élevée) pour 1.815 substantifs de la langue française (en réalité, 1.130 mots différents).

La technique de construction de ce type de dictionnaire a été reprise de celle utilisée par Paivio, Yuille, et Madigan (1968). Le dictionnaire DIMAB est appelé sous ce nom par les programmes. Cette technique consiste à demander à des juges (N=106 dans ce cas) d'indiquer sur une échelle à 7 degrés d'intensité dans quelle mesure un mot évoque peu d'images (valeur 1 sur l'échelle) ou beaucoup d'images (valeur 7 sur l'échelle). La moyenne des jugements des juges pour un mot constitue la "norme" qui est ensuite transformée en valeur de catégorie de dictionnaire. On trouve ainsi que les mots "effet" et "théorie" appartiennent à la catégorie 19 (imagerie faible) (norme issue de la valeur moyenne 1.9 sur l'échelle de 1 à 7) et que les mots "cha3teau" et "montagne" appartiennent à la catégorie 63 (imagerie élevée) (norme issue de la valeur moyenne 6.3 sur l'échelle de 1 à 7).

2. Le dictionnaire **DABS** a été construit par Hogenraad et Orianne (1981) dans la foulée du dictionnaire DIMAB présenté ci-dessus. Il s'agit de normes d'abstraction de 832 noms communs de la langue française (450 mots différents); ces normes s'étalent de la catégorie 1 (mots abstraits) à la catégorie 7 (mots concrets).

Ce petit dictionnaire a été élaboré, comme le DIMAB, sur le modèle des normes de Paivio, Yuille, et Madigan (1968). Vingt-quatre juges ont indiqué sur une échelle à 7 degrés d'intensité la mesure dans laquelle un mot est abstrait (cote 1 sur l'échelle) ou concret (cote 7 sur l'échelle). Les résultats révèlent notamment une corrélation de .90 ($t = 43.7$, $df = 448$, $p < .01$) entre les valeurs moyennes d'imagerie et d'abstraction. Le dictionnaire DABS est appelé sous ce nom par les programmes.

3. Le dictionnaire **DIRE** est, lui, un dictionnaire de catégories, en l'occurrence les catégories de processus primaires et de processus secondaires. Le DIRE est particulièrement indiqué pour l'évaluation des processus régressifs orientés vers les choses plutôt que vers les relations logiques ou symboliques (une distinction proche de celle qui existe entre pensée concrète et pensée abstraite). Ce dictionnaire a été traduit et adapté par Hogenraad de la version anglaise originale, le "Regressive Imagery Dictionary" (abrégé en RID), élaboré et validé pour cette langue par Martindale (1975; 1979).

Ce dictionnaire DIRE est composé de 3.540 mots et radicaux, répartis en 29 catégories de processus primaires, 7 catégories d'émotions, et 7 catégories de processus secondaires. Les 29 catégories de processus primaires sont elles-mêmes hiérarchisées en 5 catégories de niveau supérieur. A la différence du dictionnaire DIMAB constitué seulement de substantifs, le DIRE est constitué de verbes, adjectifs, adverbes, et substantifs; il utilise en outre intensivement les radicaux des mots. Les

deux dictionnaires DIMAB et DIRE ont été validés pour la langue française par Christine van Eeckhoudt (1981). Le dictionnaire DIRE est appelé sous ce nom par les programmes.

4. Le dictionnaire d'**agressivité** est, lui, un dictionnaire basé sur des normes d'agressivité; ces normes sont sous-divisées en quatre axes, qui sont l'agressivité simple (jugement de 1 à 7), l'agressivité physique (jugement 1) ou morale (jugement 7), l'agressivité acceptable (jugement 1) ou inacceptable (jugement 7), enfin l'agressivité interpersonnelle (jugement 1) ou institutionnelle (jugement 7). Ce dictionnaire, élaboré et validé par Jean-Paul Hostier (1987) est composé de 2.500 mots, chaque mot ayant été évalué par au moins 30 juges. L'ensemble de ce dictionnaire est appelé par le nom **AGR**, chaque axe étant appelé par l'option SUBDICT = '**AGRagr**' (pour l'intensité de l'agressivité), SUBDICT = '**AGRphy**' (pour l'axe physique-moral), SUBDICT = '**AGRacc**' (pour l'axe acceptable-inacceptable), et par SUBDICT = '**AGRins**' (pour l'axe institutionnel-interpersonnel).

5. On dispose aussi d'un dictionnaire de normes d'**évaluation, d'activation, et d'émotionnalité** de 3.000 mots (5.744 entrées). Comme pour le dictionnaire d'agressivité, ces normes reposent sur les évaluations de juges, chaque mot ayant été évalué par au moins 30 juges. L'axe d'évaluation renvoie au caractère plaisant (jugement 7) ou déplaisant (jugement 1) des mots, toujours sur une échelle à 7 degrés d'intensité. L'axe d'activation renvoie au pouvoir des mots à suggérer ou déclencher une activité (jugement 7) ou non (jugement 1). Enfin, l'axe d'émotionnalité est une estimation du degré d'association des mots à une émotion quelconque (association élevée, jugement 7). Ce dictionnaire a été élaboré et validé par Stéphane Leleu (1987). L'ensemble du dictionnaire est appelé par le nom **UCL**, l'axe d'évaluation étant appelé par les programmes via l'option SUBDICT = '**UCLev**', l'axe d'activation, via l'option SUBDICT = '**UCLac**', et l'axe d'émotionnalité, via l'option SUBDICT = '**UCLem**'.

6. Le dictionnaire de **rôles** est un dictionnaire de catégories de rôles masculins et féminins. Il a été traduit, adapté, et partiellement validé, par Christine Delcroix (1986) d'une partie du Regressive Imagery Dictionary, de Martindale (1975; 1979). Ce dictionnaire est composé de 1.396 mots et radicaux de mots (2.093 entrées) répartis en 6 catégories de thèmes masculins et 7 catégories de thèmes féminins. Il est appelé par les programmes sous le nom de **UCLROLES**.

7. Le dictionnaire **SUICIDE** repose sur une enquête au cours de laquelle 100 juges ont eu à estimer, sur une échelle à 7 points d'intensité, le degré de suicidalité (verbale) de 997 noms communs de la langue française parlée. Ce dictionnaire de normes d'imagerie suicidaire, élaboré par Monsieur Alain Descamps (1982), est composé de 969 entrées. Il est appelé sous le nom **SUICIDE** par les programmes.

8. Le "**DPSH**", **Dictionnaire Psycho-sociologique de Harvard**, a été adapté à la langue française par MM. Etienne Bocquet et Robert Hogenraad. Ce DPSH est composé de 155 catégories, les unes, exclusives, dites de premier ordre, et les autres, non-exclusives, dites de second ordre. Ensemble, ces 155 catégo-

ries constituent un dictionnaire volumineux de 6.263 entrées couvrant pratiquement tout le domaine des activités humaines étudiées par les sciences sociales.

9. Le dictionnaire "**OTTAWA**" est le dictionnaires de normes d'imagerie et de fréquence subjective de 1.916 substantifs de la langue française (Québec) de Desrochers et Bergeron (1992). Ces valeurs de fréquence et d'imagerie ont été obtenues par estimations subjectives sur des échelles de 7 degrés d'intensité, allant de peu fréquent ou imagerie faible (degré 1) à très fréquent ou imagerie élevée (degré 7). Seul les valeurs d'imagerie sont disponibles actuellement dans PROTAN.

On dispose également de dictionnaires en langue anglaise, en fait, souvent des versions anglaises, originales ou non, des dictionnaires qu'on vient de passer en revue pour la langue française.

10. Les sous-dictionnaires **IMAG**, **CONC**, et **MEAN** sont les normes de Paivio, Yuille, et Madigan (1968) mises en format de dictionnaire dans des catégories étalées de 1 à 7, au moins pour les deux premiers dictionnaires IMAG et CONC, à partir d'une échelle d'intensité de 1 à 7. Il s'agit plus précisément de normes d'imagerie (mots évoquant peu --jugement 1-- ou beaucoup --jugement 7-- d'images), d'abstraction (mots abstraits --jugement 1-- ou mots concrets --jugement 7--), et de significativité (mots susceptibles d'évoquer peu ou beaucoup d'associations, soit une échelle de 1 à n plutôt qu'une échelle de 1 à 7). Les valeurs publiées par les auteurs portent sur 925 substantifs; les sous-dictionnaires IMAG, CONC, et MEAN portent chaque fois sur les mêmes 1.508 substantifs (1.508 plutôt que 925 pour tenir compte des pluriels). L'ensemble de ces sous-dictionnaires est appelé par le nom **PAIVIO**, et les sous-dictionnaires par les options SUBDICT = '**IMAG**', SUBDICT = '**CONC**', et SUBDICT = '**MEAN**'.

11. Le dictionnaire **RID** est la version originale américaine du DIRE, le "**Regressive Imagery Dictionary**", élaboré et validé par Colin Martindale (1975; 1979). Le RID est composé de 3.978 mots répartis en 29 catégories de processus primaires, 7 catégories de processus secondaires, 7 catégories d'émotions --ces dernières théoriquement intermédiaires entre les premiers et les seconds--, 6 catégories de rôles masculins et 7 catégories de rôles féminins. Le RID est constitué de verbes, adjectifs, adverbess, et substantifs, et utilise les radicaux des mots de préférence à leurs formes libres ou nominales.

A la différence du DIRE, le RID ne comporte pas de "mots à exclure" qui ont pour but de limiter l'effet parfois non-désirable de certaines radicaux. Par exemple, le radical "abandonn." repèrera dans un texte aussi bien les mots "abandonner", qu'on souhaite inclure dans le comptage, que le mot "abandonnique" qu'on ne souhaite pas y inclure. Comme on l'a déjà indiqué dans les chapitres précédents, on obtient un effet d'exclusion en doublant le radical qu'on veut garder d'un ou plusieurs "mots à exclure" ou "radicaux à exclure"; ceux-ci se repèrent au fait qu'ils ne comportent pas d'indication de catégorie dans les colonnes 23 et suivantes. Dans l'exemple cité, les mots non souhaités "abandonnique(s)" seront exclus par

la présence du radical "abandonnique." sans indication de catégorie.

12. La série des sous-dictionnaires **UMev**, **UMac**, **UMdp**, **UMuc**, et **UMem** porte sur 558 adjectifs de langue anglaise ("UM" pour University of Massachusetts) (Averill, 1975). Ces 558 adjectifs ont été évalués par 1.600 juges (étudiants) sur 16 échelles du différenciateur sémantique d'Osgood (Osgood, Suci, & Tannenbaum, 1957), chaque juge n'évaluant qu'une partie des 558 adjectifs; chaque adjectif a pu être évalué par environ 50 à 60 juges. Les 4 axes d'**E**valuation, **A**ctivation, **P**rofondeur (DePth), et **A**bsence de Contrôle (UnControl) sont les facteurs issus de l'analyse multivariée des 16 échelles. Les scores factoriels transformés des 558 adjectifs sur les 4 facteurs constituent les normes à partir desquelles on a constitué les 4 premiers sous-dictionnaires; à ceux-ci, l'auteur a ajouté le score moyen de chaque adjectif en **émotionnalité**; ce score moyen d'émotionnalité forme le cinquième sous-dictionnaire UMem. L'ensemble de ces sous-dictionnaires est appelé par le nom AVERILL, chaque sous-dictionnaire étant appelé par les programmes via le nom qui lui est attribué ci-dessus, soit SUBDICT = '**UMev**', SUBDICT = '**UMac**', SUBDICT = '**UMdp**', SUBDICT = '**UMuc**', ou SUBDICT = '**UMem**'.

13. Les deux sous-dictionnaires **LUev** et **LUar** sont élaborés à partir de normes d'**évaluation** et d'**activation** ("arousal") de 4.000 mots de la langue anglaise. Ces normes ont été établies sur les évaluations de juges, chaque mot ayant été jugé par au moins 4 juges ("LU" pour Laurentian University) (Sweeney & Whissell, 1984; Whissell & Charuk, 1985). L'axe d'évaluation renvoie au caractère plaisant ou déplaisant des mots, et l'axe d'activation, à leur pouvoir d'évoquer l'activité, chaque fois sur une échelle à 7 degrés d'intensité (plaisant ou actif, degré 1, déplaisant ou inactif, degré 7). Les valeurs d'évaluation et d'activation des mots ont été validées par de nombreux travaux de Whissell et ses collaborateurs (Whissell, Fournier, Pellard, Weir, & Makarec, 1986); dans les deux sous-dictionnaires LUev et LUar, ces valeurs sont des valeurs transformées, où chaque axe a une moyenne de 4 et un sigma de 1. L'ensemble de ces deux sous-dictionnaires est appelé via le nom générique WHISSELL, et chaque sous-dictionnaire via l'option SUBDICT = '**LUev**' ou SUBDICT = '**LUar**'.

14. Le "**Third Harvard Psychosociological Dictionary**" est un dictionnaire de langue anglaise de 105 catégories (dont 83 catégories de base) et 3.223 mots et racines. Elaboré par Stone, Dunphy, Smith, et Ogilvie (1966), il a été l'un des premiers dictionnaires à être utilisé dans un système d'analyse du contenu assistée par ordinateur, le GENERAL INQUIRER. Ces 105 catégories représentent diverses théories sociologiques et psychologiques (catégories de rôles, d'objets, et de produits culturels et catégories de processus psychologiques dynamiques), notamment une tentative de formalisation d'une théorie générale de l'action (ibid, 1966, p. 171).

Ce dictionnaire est appelé sous le nom de HARVARD par les programmes.

15. *Note concernant la version allemande du Dictionnaire d'Imagerie Régressive.*

On dispose enfin d'une version allemande du RID de Martindale. Cette version a été élaborée par les soins de Ms. Delphend-

hal, de l'Université du Maine (U.S.A.). Cette version est appelée via le nom GGRID.

16. *Note concernant une version portugaise de normes de valeurs affectives et pro-sociales.*

Pour le portugais, on dispose à ce jour (septembre 1992) d'un dictionnaire des **valeurs affectives et pro-sociales** pour 817 mots (B. Detry, T. Lopo, & R. Hogenraad). Ce dictionnaire est appelé par le nom "**PVAPS**" et les sous-dictionnaires de valeurs affectives et pro-sociales, par les noms SUBDICT = '**PVa**', et SUBDICT = '**PVps**'.

La valeur affective des 817 mots a été obtenue à partir des jugements de 60 juges sur une échelle à 7 degrés d'intensité allant de 1 (déplaisant) à 7 (plaisant) (subdict PVA). La valeur pro-sociale des 817 mots a été obtenue par les mêmes 60 juges sur une échelle à 7 degrés d'intensité allant de 1 (valeur pro-sociale faible) à 7 (valeur pro-sociale élevée) (subdict PVps).

17. Dans la catégorie des dictionnaires de type "FINDS", on ne dispose jusqu'à présent que du seul dictionnaire "FRSTRP01 FINDS" composé, pour la langue française, de 14 mots généralement considérés comme ambigus dans la plupart des corpus.

B. Les jeux de caractères dans PROTAN

1. Caractères composants d'un mot:

+ & \$ * % _ @ A à Z et 0 à 9

2. Caractères séparateurs ordinaires:

le blanc ~ < () le non logique ¬ / , > ' = "

3. Caractères séparateurs de phrases (par défaut):

. ! ; ? :

4. Caractère encadrant un marqueur:

#

5. Caractère hybride: Composant d'un mot si simple, séparateur ordinaire si multiple:

-

```

                                finds
CFCHECKS))))))0))))))Q   CFLISTA
                                *
                                *
                                R
CSJOIN
+Q      S,s
*        *o
*        *u
/Q CSSORTS1r
*        *c
*        *e
s o) CSCHECK
u))0Q      S0)))))))3)))))))3))))))Q
r *          *
c .)))))))-
e           *       *
           *       *
           .Q      S-
             CSEDIT

                                +QCFWKWIC
w*      .    +pu
o*      .
r/QCWEDIT S,
d*      +pu *
s*      *
S)))))))3)))))))3)))))))3)))))))Q.....(suite ...)
words
+pu
/Q CWADD S1
*        *
*        *
.QCRWSTRIPS-
T    +pu
*
*
CRCHECK      S))))2))))))Q   CRLISTA
roots
```

```

      dicts
CDCHECK S))))))0))))Q CDLISTA
      *
      /))Q CDLISTC
      *
      *
      +))))))2))))),
      *          *
      R          R
CDWJUXT S))))0))Q CDWLOOK +pu
      +pu
      *
      /Q  CWKWIC
      *
      *
words
....S))))))3Q  CWKWOC +pu
      *
      *
CWORD      S3Q  CWWCOL
      *
      *
CWPAT      S3Q  CWTALLY
      *
      *
CWFLOW      S3Q  CWNEW +pu
      +pu
      *
      *
CWREFER      S2Q  CWSELECT +pu
      +pu
      T
      *
      /Q  CPEXCOR +pu
      *
      *
      .Q  CPFACOR

```

D. Condensé de la syntaxe du fichier de données SOURCE

```

colonne 11 8
123456789.....67.....0
Enregistrement 1. *SOURCE-|_____|
                    ident.      commentaire libre

ident.(ification): à partir de colonne 9, jeux 1 et 5

Enregistrement(s) 2. Commentaires libres en colonnes 1 à 80, de 0 à 999 lignes.

colonne 8
12345678.....0
Enregistrement 3. *ENDCOM~

colonne 77777777778
123456.....01234567890
Enregistrement(s) 4. |_____~ ~ ~____~____
                      texte (jeux 1 à 5)      xxx   yyyz

xxx = Interview, de 1 à 999, en format libre
yyy = Unité, de 1 à 999, en format libre
z   = Locuteur (S), jeux 1 à 5
absence des trois champs: commentaire dans le texte

```

E. Condensé de la syntaxe du fichier de données INROOT

```

                                colonnes      11                                8
                                123456789.....67.....0
Enregistrement 1.  *INROOT-|_____|_____|
                                ident.      commentaire libre
                                ident.(ification): à partir de colonne 9, jeux 1
                                et 5.
Enregistrement(s) 2. Commentaires libres en colonnes 1 à 80, de 0 à 999
                                lignes.
                                colonnes      8
                                12345678.....0
Enregistrement 3.  *ENDCOM~
                                colonnes      22222      44444      8
                                123456.....01234.....34567.....0
Enregistrement(s) 4. |_____|~~~|_____|~~~|_____|
                        ancien mot      nouveau mot      commentaire
                        ou rien

```

F. Condensé de la syntaxe du fichier de données INDICT pour dictionnaires de catégories

[illegible]

G. Condensé de la syntaxe du fichier de données INDICT pour dictionnaires de normes

```

colonne 1      11      8
1234567890....67.....0
Enregistrement 1.  *INDICN=|_____|_____|

                    ident.      commentaire libre

                    ident.(ification): à partir de colonne 9, jeux 1
                    et 5
                    (INDICN = norm-based dictionary)

Enregistrement(s) 2.  Commentaires libres en colonnes 1 à 80, de 0 à 999
                    lignes.

                    colonnes      8
                    12345678.....0
Enregistrement 3.  *ENDCOM~

                    colonnes 1      11      8
                    1234567890....67.....0
Enregistrement(s) 4.  ~~~~|_____|_____|

                    <<< >>> ident.      commentaire libre
                    (<<< = borne inférieure, >>> = borne supérieure,
                    pour chaque sous-dictionnaire, avec un maximum de
                    10; <<< toujours supérieur à 0, >>> toujours
                    inférieur à 256)

                    colonnes      22222223...
                    1234.....34567890...
Enregistrements 5.  ~~~~|_____|ccccCCCCcCCC

                    mot & racine      autant d'assignations de
                    "catégorie" qu'il y a de
                    sous-dictionnaires définis

```

H. Condensé de la syntaxe du fichier de données INFIND

```

                                colonnes      11                                8
                                123456789.....67.....0
Enregistrement 1.  *INFIND-|_____|_____|
                                ident.      commentaire libre
                                ident.(ification): à partir de colonne 9, jeux 1
                                et 5.
Enregistrement(s) 2. Commentaires libres en colonnes 1 à 80, de 0 à 999
                                lignes.
                                colonnes      8
                                12345678.....0
Enregistrement 3.  *ENDCOM~
                                colonnes      22222      44      8
                                123456.....01234.....34.....0
Enregistrement(s) 4. |_____|~~~|_____|_____|
                                mot ou racine      commentaire      commentaire
                                mémorisé      non mémorisé
                                9
                                9
                                Note: Tiret ("-") en col. 21 si mot
                                ou racine à exclure.

```

XXXV. Abrégés

Le programme CDCHECK

- Fichier de type "CDCHECK" (options)
 - LIST1 = 0, = défaut
= 1, = impression relative à la phase 1.
 - LIST2 = 0, = défaut
= 1, = impression relative à la phase 2.
 - DICT = 0,
= 1; = défaut = constitution d'un fichier de type "DICTS" pour:
 - CDLISTA
 - CDLISTC
 - CDWJUXT
 - CDWLOOK
 - SINDICT = '*'; cette instruction renvoie à une liste de (chemin et) noms de fichiers INDICT après les options (à partir d'une autre ligne et sans apostrophes).
= 'chemin et nom de fichier'; cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type INDICT.
 - Pas pour VM/CMS.
 - SDICTS = 'chemin et nom du fichier devant contenir le dictionnaire à créer' (fichier "DICTS").
 - Pas pour VM/CMS.
 - STEMP = 'c:...\'*.tmp' (défaut, STEMP = '*tmp')
- I/O:
 - I: premier fichier (partiel) de type "INDICT"
 - (éventuellement): deuxième fichier (partiel) de type INDICT
 - etc ...
 - O: fichier de type "DICTS"

Le programme CDLISTA

- Fichier de type "CDLISTA" (options)
 - SDICTS = 'chemin et nom du fichier de type DICTS'. Pas pour VM/CMS.
- I/O:
 - I: fichier de type "DICTS".
 - O: néant.

Le programme CDLISTC

- Fichier de type "CDLISTC" (options)
 - SDICTS = 'chemin et nom du fichier de type DICTS'.
 - Pas pour VM/CMS.
 - SUBDICT = Option obligatoire
 pour choisir un
 sous d'un
 (sous-dictionnaire
 entre ' et ').
- I/O:
 - I: - fichier de type "DICTS".
 - O: i. néant

Le programme CDWJUXT

(Note: L'option "NARR = 1" doit avoir été fournie au programme CSCUT).

- Fichier de type "CDWJUXT" (paramètres et options)
 - STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences) issues de CSCUT, CRWSTRIP, ou CWEDIT (associées au fichier de type "WORDS")
 - CONTEXT = 0, = défaut
= 1, = impression des phrases pour lesquelles il y a une juxtaposition de catégories des deux dictionnaires ou ensembles de catégories définis par la requête qui suit les options
= 2, = comme CONTEXT = 1, sauf que le comptage de la séquence ne porte que sur les mots du corpus (à l'exception des séparateurs, marqueurs, ponctuations, etc.) en concordance avec le programme CSCUT (option PRINT = 2)
 - LL = n,
longueur de ligne pour CONTEXT
- n peut varier de 36 à 132, 80 est le défaut
 - PUNCH = 0, = défaut
= 1, = "perforation" des statistiques par segment, fichier de type "PUNCHJ"
 - SDICTS1 = 'chemin et nom du premier fichier de type DICTS'.
■ Option pour DOS et UNIX.
 - SUBDICT1 = Option obligatoire pour choisir un sous-dictionnaire de normes s'il y en a plus d'un (sous-dictionnaire entre ' et '). SUBDICT1 se rapporte à SDICTS1.
 - SDICTS2 = 'chemin et nom du second fichier de type DICTS'.
■ Pas pour VM/CMS.
 - SUBDICT2 = Option obligatoire pour choisir un sous-dictionnaire de normes s'il y en a plus d'un (sous-dictionnaire entre ' et '). SUBDICT2 se rapporte à SDICTS2
 - SWORDS = 'chemin et nom du fichier de type WORDS'.
■ Pas pour VM/CMS.
 - SPUNCH = 'chemin et nom du fichier de type PUNCH'.
■ Pas pour VM/CMS.
 - STEMP = 'c:...\'*.tmp' (défaut, STEMP = '*.tmp')

Ensuite, une requête définie comme suit:

○ en format libre, sur une ou plusieurs lignes, les numéros de catégories des deux (sous-) dictionnaires ou ensembles de catégories d'un même dictionnaire séparés par un ou plusieurs blancs (~) (éventuellement en série caractérisée par la borne inférieure, un tiret '-', et la borne supérieure). Les deux ensembles doivent être séparés par une barre oblique ('/'). Dans le cas de deux (sous-) dictionnaires, les catégories de SDICTS1 (SUBDICT1) se mettent à gauche de la barre oblique.

● **I/O:**

- **I:** - fichier de type 'WORDS'
 - premier fichier de type 'DICTS'
 - second fichier de type 'DICTS' si deux dictionnaires différents sont utilisés
- **O:** fichier de type "PUNCHJ"

Le programme CDWLOOK

- Fichier de type "CDWLOOK" (paramètres et options)
 - STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences) issues de CSCUT, CRWSTRIP, ou CWEDIT (associées au fichier de type "WORDS")
 - WORDT = 0, = défaut
= n, = n catégories à imprimer lors de la recherche des mots
= 255, = toutes catégories à imprimer lors de la recherche des mots
 - 0 = aucune catégorie
 - n = à détailler en fin de fichier
 - 255 = toutes catégories
 - CATLOW = n
n < ou = à la catégorie la plus basse du dictionnaire, prend cette dernière valeur, = défaut;
n > à la catégorie la plus basse du dictionnaire:
 - commencer les calculs à partir de cette borne inférieure.

CATLOW, en combinaison avec PRINTT, CONT, PRINTNA ou PUNCH, n'est utilisé que pour les dictionnaires de normes (INDICN).
 - CATUP = n
n > ou = à la catégorie la plus haute du dictionnaire, prend cette dernière valeur, = défaut;
n < à la catégorie la plus haute du dictionnaire:
 - limiter les calculs à cette borne supérieure.

CATUP, en combinaison avec PRINTT, CONT, PRINTNA ou PUNCH, n'est utilisé que pour les dictionnaires de normes (INDICN)
 - SCORE = 'AVERTAG', = défaut,
= 'AVERTOT',

SCORE, en combinaison avec PRINTT, CONT, ou PUNCH, n'est utilisé que pour les dictionnaires de normes (INDICN).
 - CATSEL= 0, = défaut
= n, = n catégories à analyser et à "perforer"
= 255, = toutes catégories à analyser et à "perforer"
 - 0 = aucune catégorie
 - n = à détailler en fin de fichier

- 255 = toutes catégories

CATSEL, en combinaison avec PRINTT, CONT, PRINTNA ou PUNCH, n'est utilisé que pour les dictionnaires de catégories (INDICC).

- PRINTT= 0, = défaut
= 1, = impression des fréquences et densités par segment + scores moyens si INDICN.
- CONT = 0, = défaut
= 1, = impression de X, moyenne, écart-type et médiane par catégorie (analyse)
- WORDNT= 0, = défaut
= n, = impression de la liste des mots non repris par le dictionnaire; la sélection est opérée sur la base des n premiers caractères de ces mots
- PRINTNA= 0, = défaut
= 1, = impression du corpus dans son ordre original séquentiel; cette impression a une largeur de ligne LL et couvre le corpus du mot retenu TAGLOW au mot retenu TAGUP, du numéro de segment SEGLOW au numéro de segment SEGUP, et du numéro de séquence SEQLOW au numéro de séquence SEQUP;
= 2, = comme "PRINTNA = 1" sauf que le comptage de la séquence ne porte que sur les mots du corpus (à l'exception des séparateurs, marqueurs, ponctuations, etc) en concordance avec "PUNCH = 3 et 4" et en concordance aussi avec le programme CSCUT (option PRINT = 2).

L'option "NARR = 1" doit avoir été fournie au programme CSCUT.

- LL = n,
longueur de ligne pour PRINTNA
n peut varier de 45 à 132, 80 est le défaut
- TAGLOW = n,
borne inférieure du numéro de séquence des mots retenus pour PRINTNA, 0 est le défaut (début du corpus)
- TAGUP = n,
borne supérieure du numéro de séquence des mots retenus pour PRINTNA, le plus grand nombre de 8 chiffres est le défaut (fin du corpus)
- SEGLOW = n,
borne inférieure du numéro de segment pour PRINTNA, 1 est le défaut (début du corpus)
- SEGUP = n,
borne supérieure du numéro de segment pour PRINTNA, le plus grand nombre de 5 chiffres est le défaut (fin du corpus)
- SEQLOW = n,

borne inférieure du numéro de séquence des mots du texte pour PRINTNA, 1 est le défaut (début du corpus)

- SEQUP = n,
borne supérieure du numéro de séquence des mots du texte pour PRINTNA, le plus grand nombre de 8 chiffres est le défaut (fin du corpus)
- PUNCH = 0, = défaut
= 1, = "perforation" des statistiques par segment, fichier de type "PUNCHM", "SASCAT", ou "SASNOR"
= 2, = "perforation" des statistiques par catégorie, fichier de type "PUNCHM" ou "SASCATOB"
= 3, = "perforation" en mode séquentiel, fichier de type "PUNCHM", "SASCATSQ" ou "SASNORSQ"; l'option "NARR = 1" doit avoir été fournie au programme CSCUT
= 4; = "perforation" en mode narratif, fichier de type "PUNCHM", "SASCATNA" ou "SASNORNA"; l'option "NARR = 1" doit avoir été fournie au programme CSCUT
- SDICTS = 'chemin et nom du fichier de type DICTS'. Pas pour VM/CMS.
- SUBDICT = Option obligatoire pour choisir un sous-dictionnaire de normes s'il y en a plus d'un (sous-dictionnaire entre ' et ').
- SWORDS = 'chemin et nom du fichier de type WORDS'. Pas pour VM/CMS.
- SPUNCH = 'chemin et nom du fichier de type PUNCH'. Pas pour VM/CMS.

Ensuite, si WORDT est >0 et <255, les n catégories à considérer dans l'impression, séparées par un ou plusieurs blancs (~). Enfin, si CATSEL est >0 et <255, les n catégories à considérer dans l'analyse ou la "perforation", séparées par un ou plusieurs blancs (~).

- Fichier de type "DICTS" issu de CDCHECK
- Fichier de type "WORDS" issu de CSCUT, CRWSTRIP, CWEDIT, etc.

Appel de la procédure "CDWLOOK":

```
... CDWLOOK fn1 fn2 fn3
```

où:

- fn1 : filename du fichier de filetype "CDWLOOK", "PUNCHM", "SASCAT", "SASNOR", "SASCATOB",

"SASCATSQ", "SASNORSQ",
"SASCATNA" et "SASNORNA"

- o fn2 : filename du fichier de filetype "DICTS"
- o fn3 : filename du fichier de filetype "WORDS"

Le programme CFCHECK

- Fichier de type "CFCHECK" (options)
 - LIST1 = 0, = défaut
= 1, = impression relative à la phase 1.
 - LIST2 = 0, = défaut
= 1, = impression relative à la phase 2.
 - FIND = 0,
= 1; = défaut = constitution d'un fichier de type "FINDS" pour:
 - CFLISTA
 - CFWKWIC
 - SINFIND = '*' (et cette instruction renvoie à une liste de fichiers INFIND --sous la forme fn.ft.fm-- après les options ,à partir d'une autre ligne).
= 'fn.ft.fm' (et cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type INFIND).
Par défaut, les fichiers doivent être détaillés dans l'appel de la procédure (cfr. infra).
 - SFINDS = 'chemin et nom du fichier "DICTS" à créer'. Pas pour VM/CMS.
- Fichier(s) de type "INFIND" (données) (de 1 à 99 fichiers partiels). A noter que tous les types d'items peuvent être spécifiés, par exemple un marqueur, un signe de ponctuation donné, etc, et pas seulement des mots.

Appel de la procédure "CFCHECK":

```
... CFCHECK fn1 fn2 [fn3...]
```

où:

- fn1 : filename du fichier de filetype "CFCHECK" et "FINDS"
- fn2 : filename du premier fichier (partiel) de filetype "INFIND"
- fn3 : filename du deuxième fichier (partiel) de filetype "INFIND"
- etc...

Le programme CFLISTA

- Fichier de type "CFLISTA" (paramètres et options)
 - SFINDS ='chemin et nom du fichier de type FINDS'. Pas pour VM/CMS.
- Fichier de type "FINDS"

Appel de la procédure "CFLISTA":

```
... CFLISTA fn1 fn2
```

où:

- fn1 : filename du fichier de filetype "CFLISTA".
- fn2 : filename du fichier de filetype "FINDS".

Le programme CFWKWIC

(**Note:** L'option "NARR = 1" doit avoir été fournie au programme CSCUT).

- Fichier de type "CFWKWIC" (options)
 - STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences) issues de CSCUT, de CWEDIT, ou de CRWSTRIP (associées au fichier de type "WORDS").
 - PRINT = 0,
= 1, = défaut = impression des mots sélectionnés dans leur contexte, en ordre alphabétique.
= 2, = impression des mots sélectionnés dans leur contexte, en ordre séquentiel.
 - PUNCH = 0,
= 1, = défaut = "perforation" des mots sélectionnés, en ordre alphabétique, fichier de type "CWEDIT".
= 2, = "perforation" des mots sélectionnés, en ordre séquentiel, fichier de type "CWEDIT".
 - ANTE = n, = nombre d'items devant figurer devant le mot sélectionné; 10 est la valeur par défaut de ANTE = n.
 - POST = n, = nombre d'items devant figurer derrière le mot sélectionné; 5 est la valeur par défaut de POST = n.
 - SFINDS = 'chemin et nom du fichier de type FINDS'. Pas pour VM/CMS.
 - SWORDS = 'chemin et nom du fichier de type WORDS'. Pas pour VM/CMS.
 - SPUNCH = 'chemin et nom du fichier de type PUNCH'. Pas pour VM/CMS.
- Fichier de type "FINDS" issu de CFCHECK
- Fichier de type "WORDS" issu de CSCUT, de CWEDIT, ou de CRWSTRIP

Appel de la procédure "CFWKWIC":

... CFWKWIC fn1 fn2 fn3

où:

- fn1 : filename du fichier de filetype "CFWKWIC" (en entrée), "CWEDIT" (en sortie)
- fn2 : filename du fichier de filetype "FINDS"
- fn3 : filename du fichier de filetype "WORDS"

Le programme CRCHECK

- Fichier de type "CRCHECK" (options)
 - LIST1 = 0, = défaut
= 1, = impression relative à la phase 1.
 - LIST2 = 0, = défaut
= 1, = impression relative à la phase 2.
 - CROSS = 0, = défaut
= 1, = test des équivalences croisées avec diagnostics d'avertissement ("warning").
= 2, = test des équivalences croisées avec diagnostics d'avertissement ("warning") et informations.
 - ROOT = 0,
= 1; = défaut = constitution d'un fichier de type "ROOTS" pour CRLISTA et CRWSTRIP.
 - SINROOT = '*' (et cette instruction renvoie à une liste de fichiers INROOT --sous la forme fn.ft.fm-- après les options, à partir d'une autre ligne).
= 'fn.ft.fm' (et cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type INROOT).
Par défaut, les fichiers doivent être détaillés dans l'appel de la procédure (cfr. infra).
 - SROOTS = 'chemin et nom du dictionnaire à créer' (fichier "ROOTS"). Pas pour VM/CMS.
- Fichier(s) de type "INROOT" (données) (de 1 à 99 fichiers partiels). A noter que tous les types d'items peuvent être spécifiés, outre des mots, comme par exemple un marqueur ou un signe de ponctuation.

Appel de la procédure "CRCHECK":

```
... CRCHECK fn1 fn2 [fn3...]
```

où:

- fn1 : filename du fichier de filetype "CRCHECK" et "ROOTS"
- fn2 : filename du premier fichier (partiel) de filetype "INROOT"
- fn3 : filename du deuxie2me fichier (partiel) de filetype "INROOT"
- etc...

Le programme CRLISTA

- Fichier de type "CRLISTA" (options)
 - SORTT = 'OLD', = défaut = tri sur les anciens mots
= 'NEW'; = tri sur les nouveaux mots.
 - SROOTS = 'chemin et nom du dictionnaire' (fichier "ROOTS").
 - Pas pour VM/CMS
- Fichier de type "ROOTS".

Appel de la procédure "CRLISTA":

`... CRLISTA fn1 fn2`

où:

- fn1 : filename du fichier de filetype "CRLISTA"
- fn2 : filename du fichier de filetype "ROOTS"

Le programme CRWSTRIP

- Fichier de type "CRWSTRIP" (options)
 - TABLE = 0, = défaut
= 1, = impression de la table des mots triés en sortie.
 - STAT = 0,
= 1, = défaut = impression des statistiques
de tri (fréquences) en entrée et en sortie.
 - ROOTT = 0, = défaut
= 1, = impression de la chaîne des transformations
effectuées.
 - PUNCH = 0, = défaut
= 1, = "perforation" des mots triés, fichier
de type "PUNCHW" ou "SASWORD".
 - WORD = 0,
= 1, = défaut = constitution d'un fichier de
type "WORDS" pour les programmes ultérieurs.
 - SROOTS = 'chemin et nom du dictionnaire' (fichier "ROOTS").
 - Pas pour VM/CMS.
 - SWORDS = 'chemin et nom du corpus' en entrée (fichier
"WORDS").
 - Pas pour VM/CMS.
 - SWORDSO = 'chemin et nom du corpus' en sortie (fichier
"WORDS").
 - Pas pour VM/CMS.
 - SPUNCH = 'chemin et nom du fichier PUNCH' (correspondant au
fichier "WORDS" produit en sortie).
 - Pas pour VM/CMS
- Fichier de type "ROOTS" issu de CRCHECK
- Fichier de type "WORDS" issu de CSCUT, CRWSTRIP, ou CWEDIT

Appel de la procédure "CRWSTRIP":

```
... CRWSTRIP fn1 fn2 fn3
```

où:

- fn1 : filename du fichier de filetype "CRWSTRIP" (en entrée),
"WORDS", "PUNCHW" et "SASWORD" (en sortie)
- fn2 : filename du fichier de filetype "ROOTS"
- fn3 : filename du fichier de filetype "WORDS" (en entrée)

Le programme CSCHECK

- Fichier de type "CSCHECK" (options)
 - PRINT = 0, = défaut
= 1, = impression des enregistrements "source"
 - COMM = 0, = pas de commentaires (qui feraient partie du corpus)
= 1, = défaut, les commentaires sont imprimés
 - WARN = 0, = pas de messages d'avertissement de type 'W',
= 1, = les messages d'avertissement de type 'W' sont imprimés.
 - SENT = '' ou SENT = '.!;?:', = défaut
= à choisir dans les jeux de caractères 2 et 3, permettant de définir soi-même les séparateurs de phrases (sauf le caractère ~).
 - SSOURCE = '*': cette instruction renvoie à une liste de (chemin et) noms de fichiers SOURCE après les options (à partir d'une autre ligne et sans apostrophes).
= 'chemin et nom de fichier': cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type SOURCE.
- I/O:
 - I: premier fichier (partiel) de type "SOURCE" (données).
- (éventuellement): deuxième fichier (partiel) de type SOURCE
 - O: néant

Le programme CSCUT

- Fichier de type "CSCUT" (options)
 - TABLE = 0, = défaut
= 1, = impression de la table des mots triés.
 - STAT = 0,
= 1, = défaut = impression des statistiques de tri (fréquences) et autres informations sur la longueur moyenne des mots et des phrases.
 - PSTAT = 0, = défaut
= 1, = perforation des statistiques de tri (fréquences) et autres informations sur la longueur moyenne des mots et des phrases, fichier de type "PUNCHS".
 - PRINT = 0, = défaut
= 1, = impression des enregistrements "source" avec comptage des séparateurs;
= 2, = impression des enregistrements "source" sans comptage des séparateurs.
 - COMM = 0, = pas de commentaires (qui feraient partie du corpus)
= 1, = défaut, les commentaires sont imprimés
 - PUNCH = 0, = défaut
= 1, = "perforation" des mots triés, fichier de type "PUNCHW".
 - WORD = 0,
= 1, = défaut = constitution d'un fichier de type "WORDS" pour les programmes ultérieurs.
 - CTRL = n,
- doit être >0 si SEGT est utilisé.
 - SEGT = 'W'...mot,
= '#marqueur#',
= 'S'...phrase,
- type de resegmentation (optionnel).
 - RESIDUE = '**keep**' = défaut, permet de conserver telle quelle la valeur du dernier segment produit par les options MODULO ou <SEGT='w'>;
= '**drop**', supprime la valeur du dernier segment produit par MODULO ou <SEGT = 'w'>;
= '**merge**', ajoute la valeur du dernier segment créé par MODULO ou <SEGT = 'w'> à la valeur du segment précédent;
 - MODULO = 'n', vaut zéro par défaut, sinon divise chaque segment initial du corpus en un nombre strictement égal de mots;
 - REGEN = 1 ou 0 (0 par défaut); <REGEN = 1>, agit sur les nouveaux segments générés par MODULO en sorte que se trouvent réunis tous les nièmes segments générés par MODULO. Le nombre total de segments est dans ce cas égal à la valeur de MODULO.
 - SENT = '' ou SENT = ';.!?:', = défaut
= à choisir dans les jeux de caractères 2 et 3, permettant de définir soi-même les séparateurs de phrases (sauf le ~).
 - BRKM = n,

- doit être >0, défaut = 1, nombre maximum de "breaks" avant le changement de segment.

- BRKB = 'I',
= 'U',
= 'S',
= 'IU',
etc...

- frontière du "break", combinaison de 1 à 3 codes, option obligatoire.

- NARR = 0, = défaut
= 1; = confère au fichier de type "WORDS" les attributs permettant de reconstituer ce fichier dans l'ordre original séquentiel lors de l'exécution des programmes CDWLOOK et CWSELECT (options "PRINTNA = 1", "PUNCH = 3" et "PUNCH = 4"), CFWKWIC, CWEDIT, CWKWIC et CWPAT.

- SSOURCE = '*': cette instruction renvoie à une liste de (chemin et) noms de fichiers SOURCE après les options (à partir d'une autre ligne et sans apostrophes).

- = 'chemin et nom de fichier': cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type SOURCE.

- SWORDS = 'chemin et nom du corpus' en sortie (fichier "WORDS").

- SPUNCH = 'chemin et nom du fichier PUNCHW' (correspondant au fichier "WORDS" produit en sortie).

- SPSTAT = 'chemin et nom du fichier PUNCHS' (correspondant aux statistiques de tri produites en sortie).

- STEMP = 'c:...*.tmp' (défaut, STEMP = '*.tmp').

- Fichier(s) de type "SOURCE" (données), de 1 à 99 fichiers partiels.

Appel de la procédure "CSCUT":

... CSCUT fn1 fn2 [fn3...]

où:

- fn1 : filename du fichier de filetype "CSCUT" (en entrée), "WORDS", "PUNCHW", "PUNCHS" (en sortie).

- fn2 : filename du premier fichier (partiel) de filetype "SOURCE"

- fn3 : filename du deuxième fichier (partiel) de filetype "SOURCE"

- etc...

Le programme CSEDIT

- Fichier de type "CSEDIT" (paramètres et options)

Options

- PRINT = 0, = défaut
= 1, = impression des enregistrements retenus ou renumérotés
- COMM = 0, = pas de commentaires (qui feraient partie du corpus)
= 1, = défaut, les commentaires sont imprimés
- OUT = 0,
= 1, = défaut = constitution d'un fichier de type "SOURCE"
- EDIT = 'PASS',
= 'DROP',
= 'RENUM',
option obligatoire.
- FIELD = 'I',
= 'U',
= 'S',
= 'IU',
etc...
= 'IUS';
(=champ sur lequel s'applique l'édition) (option obligatoire).
- SSOURCE = '*' (et cette instruction renvoie à une liste de fichiers SOURCE --sous la forme fn.ft.fm-- après les paramètres et une ligne blanche de séparation).
= 'fn.ft.fm' (et cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type SOURCE).
Par défaut, les fichiers doivent être détaillés dans l'appel de la procédure (cfr. infra).
Ce programme accepte également des fichiers SOURCE à concaténer.
- SSOURCEO = 'chemin et nom du nouveau fichier SOURCE' créé en sortie, surtout utile pour une utilisation sur PC.

Enregistrements 1, 2 et 3 destinés à la constitution du nouveau fichier de type "SOURCE".Contenu(s) du champ spécifié par FIELD faisant l'objet de l'édition définie par EDIT, suivant les règles de syntaxe des champs 73 à 80 des enregistrements "SOURCE":

colonnes	
12345678	
xxx	...pour PASS/DROP I
xxx	...pour PASS/DROP U
x	...pour PASS/DROP S
xxx xxx	...pour PASS/DROP IU
etc...	

```
colonnes 11111111
12345678..12345678
xxx      xxx      ...pour RENUM I
xxx      x  xxx      x  ...pour RENUM IS
xxx xxx  xxx xxx  ...pour RENUM IU
etc...
```

- Fichier(s) de type "SOURCE" (données), de 1 à 99 fichiers à concaténer, pour un maximum de 99 fichiers partiels.

Appel de la procédure "CSEDIT":

```
... CSEDIT fn1 fn2 [fn3 ...]
```

où:

- fn1 : filename du fichier de filetype "CSEDIT" (en entrée)
...de filetype "SOURCE" (en sortie)
- fn2 : filename du premier fichier (partiel) de filetype "SOURCE" (en entrée)
- fn3 : filename du deuxième fichier (partiel) de filetype "SOURCE" (en entrée)
- etc...

Le programme CSJOIN

- Fichier de type "CSJOIN" (paramètres et options)

Options

- PRINT = 0, = défaut
= 1, = impression des enregistrements "SOURCE"
- COMM = 0, = pas de commentaires (qui feraient partie du corpus)
= 1, = défaut, les commentaires sont imprimés
- OUT = 0,
= 1; = défaut = constitution d'un fichier de type "SOURCE" résultant
- FILL = 1, création, dans un fichier-source, des codes d'interview, d'unité et de speaker dans les zones appropriées,
= 0, défaut, pas d'altération du fichier
- FILLER1 = la zone 73-75 est remplie de '***' (par défaut)
- FILLERU = la zone 77-79 est remplie de '%%%' (par défaut)
- FILLERS = la colonne 80 est occupée par le signe '@' (défaut)
- SSOURCE = '*' (et cette instruction renvoie à une liste de fichiers SOURCE --sous la forme fn.ft.fm-- après les paramètres).
= 'fn.ft.fm' (et cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type SOURCE).
Par défaut, les fichiers doivent être détaillés dans l'appel de la procédure (cfr. infra).
Ce programme accepte également des fichiers SOURCE à concaténer.
- SSOURCEO = 'chemin et nom du nouveau fichier SOURCE' créé en sortie, surtout utile pour une utilisation sur PC.

Enregistrements 1, 2, et 3 destinés à la constitution du nouveau fichier "SOURCE".

(Voir Chapitres IV et XXXII sur la syntaxe des fichiers de données "SOURCE").

- Fichiers de type "SOURCE" (données), de 1 à 99 fichiers à concaténer, pour un maximum de 99 fichiers partiels.

Appel de la procédure "CSJOIN":

... CSJOIN fn1 fn2 [fn3...]

où:

- fn1 : filename du fichier de filetype "CSJOIN" en entrée
...de filetype "SOURCE" en sortie
- fn2 : filename du premier fichier CMS de filetype "SOURCE"
- fn3 : filename du deuxième fichier CMS de filetype "SOURCE"
- etc...

Le programme CSSORT

- Fichier de type "CSSORT" (paramètres et options)

Options

- LIST1 = 0, = défaut
= 1, = impression de la phase 1 (données originales)
- LIST2 = 0, = défaut
= 1, = impression de la phase 2 (résultats du tri)
- COMM = 0, = pas de commentaires (qui feraient partie du corpus)
= 1, = défaut, les commentaires sont imprimés
- OUT = 0,
= 1, = défaut = constitution d'un fichier de type "SOURCE"
- SORTT = 'I',
= 'U',
= 'S',
= 'IU',
= 'UI',
= 'IUS',
= etc...
= 'SUI';
 - SORTT = type de tri
 - défaut = 'IU'
 - combinaison de 1 à 3 codes selon la hiérarchie souhaitée.
- SSOURCE = '*' (et cette instruction renvoie à une liste de fichiers SOURCE --sous la forme fn.ft.fm-- après les paramètres).
= 'fn.ft.fm' (et cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type SOURCE).
Par défaut, les fichiers doivent être détaillés dans l'appel de la procédure (cfr. infra).
Ce programme accepte également des fichiers SOURCE à concaténer.
- SSOURCEO = 'chemin et nom du nouveau fichier SOURCE' créé en sortie, surtout utile pour une utilisation sur PC.

Enregistrements 1, 2, et 3 destinés à la constitution du nouveau fichier de type "SOURCE".

- Fichier(s) de type "SOURCE" (données), de 1 à 99 fichiers à concaténer, pour un maximum de 99 fichiers partiels.

Appel de la procédure "CSSORT":

```
... CSSORT fn1 fn2 [fn3...]
```

où:

- fn1 : filename du fichier de filetype "CSSORT" (en entrée)
...de filetype "SOURCE" (en sortie)

- fn2 : filename du premier fichier (partiel) de filetype
"SOURCE" (en entrée)
- fn3 : filename du deuxième fichier (partiel) de filetype
"SOURCE" (en entrée)
- etc...

Le programme CWADD

- Fichier de type "CWADD" (paramètres et options)

Options

- PRINT = 0, = défaut
= 1, = impression des paramètres.
- STAT = 0,
= 1, = défaut = impression des statistiques de tri
(fréquences associées au fichier de type "WORDS").
- WORD = 0,
= 1, = défaut = constitution d'un fichier de type
"WORDS".
- SWORDS = 'chemin et nom du corpus' en entrée, fichier de type
"WORDS", surtout utile en PC.
- SWORDS0 = définition du fichier créé par le programme en
sortie (fichier de type "WORDS"), sous la forme 'chemin et nom du
corpus'.

Paramètres: Présentation des événements non verbaux à inclure
dans le fichier "WORDS" selon le format:

- col 1 à 20 : identification
 --doit commencer
 par ## et avoir la
 syntaxe d'un mot--;
 lorsque cette zone
 est vierge, le
 programme prend la
 valeur précédente
 non vierge.
- col 21 à 23: vierges
- col 24 à 28: numéro de segment
- col 29 à 31: vierges
- col 32 à 39: fréquence associée
- col 40 à 42: vierges
- col 43 à 80: réserve.

- Fichier de type "WORDS" issu de CSCUT, CWADD, de CWEDIT, ou de
CRWSTRIP

Appel de la procédure "CWADD":

... CWADD fn1 fn2

où:

- fn1 : filename du fichier de filetype "CWADD" (en entrée) et
"WORDS" (en sortie)
- fn2 : filename du fichier de filetype "WORDS" (en entrée)

Le programme CWEDIT

(**Note:** L'option "NARR = 1" doit avoir été fournie au programme CSCUT. Ceci dit, tous les types d'items --séparateurs, ponctuations, etc.-- peuvent être ainsi transformés par CWEDIT).

- Fichier de type "CWEDIT" (paramètres et options)

Options

- FID = identification optionnelle des paramètres (entre apostrophes) (maximum 8 caractères)
- FCOM = commentaire optionnel associé à FID (entre apostrophes) (maximum 64 caractères)
- FDATE = date optionnelle associée à FID (entre apostrophes) (maximum 8 caractères)
- FTIME = heure optionnelle associée à FID (entre apostrophes) (maximum 8 caractères)
- TABLE = 0, = défaut
= 1, = impression de la table des mots triés en sortie.
- STAT = 0,
= 1, = défaut = impression des statistiques de tri (fréquences) en entrée et en sortie.
- PRINT = 0, = défaut
= 1, = impression des paramètres.
- PUNCH = 0, = défaut
= 1, = "perforation" des mots triés, fichier de type "PUNCHW" ou "SASWORD".
- WORD = 0,
= 1, = défaut = constitution d'un fichier de type "WORDS" pour des programmes ultérieurs.
- SWORDS = 'chemin et nom du corpus' en entrée (fichier "WORDS"), surtout utile pour une utilisation sur PC.
- SWORDSO = 'chemin et nom du corpus "édité"' en sortie (fichier "WORDS"), surtout utile pour une utilisation sur PC.
- SPUNCH = 'chemin et nom du fichier PUNCH' (correspondant au fichier "WORDS" produit en sortie), surtout utile pour une utilisation sur PC.

Paramètres: Sélection des mots à remplacer ou à supprimer selon le format (un enregistrement par mot):

- col 1 à 8 : numéro de séquence du mot en format libre
- col 9 à 11 : vides
- col 12 à 31: ancien mot du corpus
- col 32 à 34: vides
- col 35 à 54: nouveau mot (remplacement)
ou vides (suppression)
- col 55 à 57: vides

-
- col 58 à 77: commentaire optionnel
 - col 78 à 80: réserve

- Fichier de type "WORDS" issu de CSCUT, de CWEDIT, ou de CRWSTRIP

Appel de la procédure "CWEDIT":

```
... CWEDIT fn1 fn2
```

où:

- fn1 : filename du fichier de filetype "CWEDIT" (en entrée), "WORDS", "PUNCHW" et "SASWORD" (en sortie)
- fn2 : filename du fichier de filetype "WORDS" (en entrée)

Le programme CWFLOW

(**Note:** L'option "NARR = 1" doit avoir été fournie au programme CSCUT).

- Fichier de type "CWFLOW" (paramètres et options):

- EXAEQUO = 0, = pas de suppression des intervalles contenant le même nombre de nouveaux mots,
= 1, = défaut = suppression des intervalles contenant le même nombre de nouveaux mots.

- NWEND = 0, = pas de suppression des intervalles qui ne se terminent pas par un nouveau mot,
= 1, = défaut = suppression des intervalles qui ne se terminent pas par un nouveau mot.

- FIELD = étendue de la "fenêtre" de recherche pour la moyenne mobile, = 55, fenêtre de 55 mots, = défaut.

- SEGNO = 0, = défaut = commande l'exécution de CWFLOW pour tout le texte, = **n**, = commande l'exécution de CWFLOW pour le segment de texte défini par la valeur de **n**,

- PRINT = 0, = défaut,
= 1, = impression du contenu des intervalles de la fenêtre mobile.

- LL = n, largeur de ligne pour le paramètre PRINT (n peut varier de 36 à 132, 80 est le défaut).

- STAT = 0, = défaut,
= 1, = impression des statistiques de tri (fréquences) issues de CSCUT, CRWSTRIP, etc (associées au fichier de type "WORDS").

- STRIP = 0, = niveau 0, fichier "WORDS" non strippé,
= 1, = défaut, niveau le plus récent du strip, CWFLOW sur fichier strippé.

- PUNCH = 0, = défaut
= 1, = "perforation" des fréquences des "types", fichier de type "PUNCHV" ou "SASFLOW".

- SPUNCH = 'chemin et nom du fichier PUNCHV' produit en sortie.
■ Pas pour VM/CMS.

- SWORDS = définition du fichier "WORDS" contenant le corpus, sous la forme 'chemin et nom du fichier WORDS'.
■ Pas pour VM/CMS.

- STEMP = 'c:...*.tmp' (défaut, STEMP = '*.tmp')

(Note: Pour le calcul des statistiques de séries temporelles (auto-corrélations, etc.), les paramètres EXAEQUO et NWEND doivent être mis à 0 tous les deux. Cette précaution prend en compte les points intermédiaires qui sont supprimés pour exécuter le tracé des types (nombre de mots différents nouveaux par rapport à tout ce qui précède).

- I/O:

- I: fichier de type "WORDS".

- O: fichier de type "PUNCHV".

Le programme CWKWIC

(**Note:** L'option "NARR = 1" doit avoir été fournie au programme CSCUT).

- Fichier de type "CWKWIC" (options et mots à localiser)
 - CONTEXT = 0, = défaut (contexte absent)
= 1 = mots apparaissent avec leur contexte
 - MAXREF= nombre maximum de mots-clés à lire; défaut = 1.000.
 - SORTW = 0, = défaut
= 1; = trier préalablement les mots-clés.
 - STAT = 0, = défaut
= 1; = impression des statistiques de tri (fréquences) issues de CSCUT, CRWSTRIP, etc (associées au fichier de type "WORDS").
 - STRIP = 0, = défaut (niveau 0, fichier "WORDS" non strippé),
= 1; = niveau le plus récent du strip, KWIC sur fichier strippé.
 - SWORDS = définition du fichier "WORDS" contenant le corpus, sous la forme 'chemin et nom du fichier WORDS', surtout utile en PC.
- Ensuite, les mots à localiser, entre apostrophes et séparés par une virgule et/ou un ou plusieurs blancs (~).
- Jeux autorisés:* tous et le "blanc" (~) pour encadrer un mot, précéder un préfixe, ou suivre un suffixe.

- Fichier de type "WORDS" issu de CSCUT ou autres.

Appel de la procédure "CWKWIC":

```
... CWKWIC fn1 fn2
```

où:

- fn1 : filename du fichier de filetype "CWKWIC"
- fn2 : filename du fichier de filetype "WORDS".

Le programme CWKWOC

- Fichier de type "CWKWOC" (options et mots-clés à localiser)
 - SORTW = 0, = défaut
= 1, = trier préalablement les mots-clés à localiser.
 - STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences) issues de CSCUT, de CWEDIT, ou de CRWSTRIP (associées au fichier de type "WORDS").
 - TABLE = 0, = défaut
= 1, = impression, par mot-clé, de la table des mots correspondants repérés dans le fichier "WORDS".
 - PUNCH = 0, = défaut
= 1, = "perforation" des mots triés, fichier de type "**PUNCHW**" ou "SASWORD".
= 2, = constitution d'un fichier de type "**INROOT**" destiné à CRCHECK et à éditer par l'utilisateur.
= 3, = constitution d'un fichier de type "**REFER**" destiné à CWSELECT et CWORD.
= 4, = constitution d'un fichier de type "**INFIND**" destiné à CFCHECK et à éditer éventuellement par l'utilisateur.

A chacune des valeurs respectives de "PUNCH =" correspond une option:

- de SPUNCH = 'chemin et nom du fichier PUNCHW' (pour PUNCH = 1),
 - de SINROOT = 'chemin ...fichier INROOT' (pour PUNCH = 2),
 - de SREFER = 'chemin ...fichier REFER' (pour PUNCH = 3),
 - de SINFIND = 'chemin ...fichier INFIND' (pour PUNCH = 4).
- MAXREF = nombre maximum de mots-clés; défaut
= 1.000, maximum = 32.000.
 - SWORDS = définition du fichier "WORDS" contenant le corpus, sous la forme 'chemin et nom du fichier WORDS', surtout utile en PC.

Ensuite, les mots-clés à localiser, entre apostrophes et séparés par une virgule et/ou un ou plusieurs blancs (~).

Jeux autorisés: 1, 5, et le "blanc" (~) pour encadrer un mot, précéder un préfixe, ou suivre un suffixe.

- Fichier de type "WORDS" issu de CSCUT, de CWEDIT, ou de CRWSTRIP

Appel de la procédure "CWKWOC":

... CWKWOC fn1 fn2

où:

- fn1 : filename du fichier de filetype "CWKWOC" (en entrée), "PUNCHW", "SASWORD", "INROOT", "REFER" et "INFIND" (en sortie)

o fn2 : filename du fichier de filetype "WORDS"

Le programme CWNEW

- Fichier de type "CWNEW" (paramètres et options)

Options

- STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences)
issues de CSCUT, CRWSTRIP, ou CWEDIT (associées aux fichiers de type "WORDS")
- TABLE = 0, = défaut
= 1, = impression de la table des mots nouveaux (NEW),
absents du groupe de mots anciens (OLD).
- PUNCH = 0, = défaut
= 1, = "perforation" des indices d'accroissement du
vocabulaire, fichier de type "PUNCHI" ou "SASINC".
- WORDNT= 0, = défaut
= n, = impression de la liste des mots non repris par
l'option TABLE; la sélection est opérée sur la base des n premiers
caractères de ces mots.
- SWORDS= '*' (et cette instruction renvoie à une liste de
fichiers WORDS --sous la forme fn.ft.fm-- après les options mais
avant les paramètres, en terminant la liste par le code * si le
nombre de ces fichiers WORDS est <9).
= 'fn.ft.fm' (et cette instruction donne le nom de
fichier unique au cas où il n'y a qu'un seul fichier de type WORDS).
Par défaut, les fichiers doivent être détaillés dans
l'appel de la procédure (cfr. infra).
- SPUNCH = 'chemin et nom du fichier PUNCHI' produit en sortie,
surtout utile pour une utilisation sur PC.
- STEMP = 'c:...*.tmp' (défaut, STEMP = '*.tmp')

Deux types de paramètres (à utiliser en alternance):

○ par ligne, une définition de segment(s) en format libre,
c'est-à-dire:

- [blanc(s) (~) optionnels]
- identification du fichier "WORDS" (voir
*SOURCE-identification)
- [un ou plusieurs blancs (~)]
- numéro du segment

ou

en série: 2 numéros séparés par un "-" pouvant être
encadré par un ou plusieurs blancs (~)

- [un ou plusieurs blancs (~)]
- l'appartenance au groupe de mots anciens
(OLD)/nouveaux (NEW) par les lettres O ou N, respectivement
- [un ou plusieurs blancs (~)]

- commentaire optionnel
 - o un mot-clé demandant le traitement en fonction des définitions qui précèdent (sur une ligne), c'est-à-dire:
 - [blancs (~) optionnels]
 - un caractère "/"
 - [blancs (~) optionnels]
 - le mot-clé RUN
 - un ou plusieurs blancs (~)]
 - commentaire optionnel (utilisé comme titre de page pour le traitement correspondant.

- Les n (de 1 à 9) fichiers de type "WORDS" contenant le matériel à traiter et correspondant aux définitions du fichier de type "CWNEW"; un fichier par identification différente fournie.

Appel de la procédure "CWNEW":

`... CWNEW fn1 fn2 [fn3...]`

où:

- o fn1 : filename du fichier de filetype "CWNEW", "PUNCHI" et "SASINC"
- o fn2 : filename du premier fichier de filetype "WORDS" (matériel à traiter)
- o fn3 : filename du deuxième fichier de filetype "WORDS" (matériel à traiter)
- o etc...

Le programme CWPAT

(**Note:** L'option "NARR = 1" doit avoir été fournie au programme CSCUT).

- Fichier de type "CWPAT" (options et paramètres)
 - CONTEXT = 0, = défaut (contexte absent)
= 1 = mots apparaissent avec leur contexte
 - MAXREF= nombre maximum de mots-clés à lire par requête;
défaut = 20.
 - MAXTAG= nombre maximum de mots indexés par tous les mots-clés réunis (par
requête); défaut = 1.000.
 - STAT = 0, = défaut
= 1; = impression des statistiques de tri (fréquences)
issues de CSCUT, CRWSTRIP, etc (associées au fichier de type "WORDS").
 - STRIP = 0, = niveau 0, fichier "WORDS" non strippé,
= 1; = défaut, niveau le plus récent du strip, CWPAT sur
fichier strippé.
 - FIELD = nombre positif = étendue de la "fenêtre" de
recherche,
= nombre négatif = recherche ne s'arrête qu'à la limite
du segment = défaut.
 - EXTEND = 0; = défaut, recherche se poursuit à partir du
dernier mot trouvé (inclus) dans le corpus,
= 1;= recherche se poursuit au n+1 mot trouvé;
 - TEXTW = 0; = pas d'impression des TEXT WORDS,
= 1; = impression des TEXT WORDS trouvés dans le corpus
avec indication des numéros de séquence et de segment = défaut.
 - SWORDS = définition du fichier "WORDS" contenant le corpus,
sous la forme 'chemin et nom du fichier WORDS', surtout utile en PC.

Ensuite, les paramètres sous forme de mots-clés, chaque requête se terminant par un / mais débutant toujours à la ligne. Un commentaire peut cependant suivre le /.

- Fichier de type "WORDS" issu de CSCUT ou autres.
 - Note sur la syntaxe des paramètres de CWPAT: De façon générale, la syntaxe des paramètres (séparateurs et codage des mots-clés) est celle d'un fichier-source. Les opérations logiques de recherche de patterns sont réalisées à l'aide de:
 - l'opération logique "**ET**" toujours implique et ayant toujours lieu à l'extérieur des parenthèses;
 - (), c'est-à-dire opération logique "OU";
 - < >, c'est-à-dire recherche à effectuer DANS L'ORDRE où les mots-clés se présentent, sinon dans le DESORDRE. Une recherche "dans l'ordre" ne peut être incluse à l'intérieur d'une paire de parenthèses;

- mots-clés ou racines de mots-clés, chaque
racine se termine par un point.

Appel de la procédure "CWPAT":

... CWPAT fn1 fn2

où:

- o fn1 : filename du fichier de filetype "CWPAT"
- o fn2 : filename du fichier de filetype "WORDS".

Le programme CWREFER

- Fichier de type "CWREFER" (options)
 - STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences)
issues de CSCUT, CRWSTRIP, ou CWEDIT (associées au fichier de type "WORDS")
 - FMIN = 0, = défaut = fixé à 2
= n, = fréquence minimale pour qu'un mot du fichier "WORDS" soit sélectionné.
 - FSEL = 'S', = défaut = FMIN s'applique à la fréquence par segment
= 'T', = FMIN s'applique à la fréquence totale
 - PUNCH = 0, = pas de perforation
= 1, = défaut = perforation des mots sélectionnés.
 - SPUNCH = définition du fichier de perforation (sortie) de type "REFER", sous la forme 'chemin et nom du fichier REFER' (surtout utile en PC).
 - SWORDS = définition du fichier contenant le corpus (entrée) de type "WORDS", sous forme 'chemin et nom de fichier' (surtout pour PC).
- Fichier de type "WORDS".

Appel de la procédure "CWREFER":

... CWREFER fn1 fn2

où:

- fn1 : filename du fichier de filetype "CWREFER" (en entrée), "REFER" (en sortie)
- fn2 : filename du fichier de filetype "WORDS"

Le programme CWSELECT

● Fichier de type "CWSELECT" (options)

- STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences)
issues de CSCUT ou de CRWSTRIP (associées au fichier de type
"WORDS").
- MAXREF = nombre maximum de mots de référence à lire; défaut =
1.000, maximum = 32.000.
- FIELD = nombre positif = étendue de la "fenêtre" de recherche
de pattern,
= nombre négatif = recherche ne s'arrête qu'à la limite
du segment = défaut.
- EXTEND = 0; = défaut, la recherche de pattern se poursuit à
partir du dernier mot trouvé (inclus) dans le corpus,
= 1; = recherche se poursuit au n+1 mot trouvé.
- MAXPAT = nombre maximum de mots-clés à lire par requête de
recherche de pattern; défaut = 20.
- MAXTAG = nombre maximum de mots indexés par tous les mots-clés
réunis (par requête); défaut = 1.000.
- SORTW = 0, = défaut
= 1, = trier préalablement les mots du fichier de type
"REFER".
- PRINT = 0, = défaut
= 1, = impression de la matrice de fréquences (absolues
et relatives).
- PUNCH = 0, = défaut
= 1, = "perforation" de la matrice des fréquences
(absolues et relatives) des "mots par segment", fichier de type
"PUNCHF" ou "SASSEL".
= 2, = "perforation" de la matrice des fréquences
(absolues et relatives) des "segments par mot", fichier de type
"PUNCHF" ou "SASSELOB".
= 3, = "perforation" en mode séquentiel, fichier de type
"PUNCHM" ou "SASSELSQ"; l'option "NARR = 1" doit avoir été fournie au
programme CSCUT
= 4; = "perforation" en mode narratif, fichier de type
"PUNCHM" ou "SASSELNA"; l'option "NARR = 1" doit avoir été fournie au
programme CSCUT. Normalement sur niveau 0 du fichier WORDS.
- PRINTNA = 0, = défaut
= 1, = impression du corpus dans son ordre original
séquentiel; cette impression a une largeur de ligne LL et couvre le
corpus du mot retenu HITLOW au mot retenu HITUP, du numéro de segment
SEGLOW au numéro de segment SEGUP, et du numéro de séquence SEQLOW au
numéro de séquence SEQU.
= 2, = comme "PRINTNA = 1" sauf que le comptage de la
séquence ne porte que sur les mots du corpus (à l'exception des
séparateurs, marqueurs, ponctuations, etc) en concordance avec "PUNCH
= 3 et 4" et en concordance aussi avec le programme CSCUT (option
PRINT = 2).
L'option "NARR = 1" doit avoir été fournie au programme
CSCUT. Sur niveau 0 du fichier WORDS.
- LL = n, largeur de ligne pour PRINTNA
n peut varier de 45 à 132, 80 est le défaut

○ HITLOW = n, borne inférieure du numéro de séquence des mots retenus pour PRINTNA,
0 est le défaut (début du corpus)

○ HITUP = n, borne supérieure du numéro de séquence des mots retenus pour PRINTNA, le plus grand nombre de 8 chiffres est le défaut (fin du corpus)

○ SEGLOW = n, borne inférieure du numéro de segment pour PRINTNA, 1 est le défaut (début du corpus)

○ SEGUP = n, borne supérieure du numéro de segment pour PRINTNA, le plus grand nombre de 5 chiffres est le défaut (fin du corpus)

○ SEQLOW = n, borne inférieure du numéro de séquence dans le corpus. Le comptage se fait en fonction de PRINTNA. "SEQLOW = 1" est le défaut (début du corpus)

○ SEQUP = n, borne supérieure du numéro de séquence dans le corpus. Le comptage se fait en fonction de PRINTNA. La valeur par défaut attribuée à SEQUP est le plus grand nombre de 8 chiffres (fin du corpus).

○ SPUNCH = définition du fichier de perforation (sortie) de type "PUNCHF", sous la forme 'chemin et nom du fichier PUNCHF' (surtout utile en PC).

○ SWORDS = définition du fichier contenant le corpus (entrée) de type "WORDS", sous forme 'chemin et nom de fichier' (surtout pour PC).

○ SREFER = définition du fichier REFER, sous la forme 'chemin et nom du fichier REFER' (surtout utile en PC).

● Fichier de type "REFER" (le fichier de référence): c'est-à-dire les différents mots de référence, entre apostrophes, et séparés par une virgule et/ou un ou plusieurs blancs (~) (maximum de MAXREF mots).

Les mots peuvent être groupés en étant précédés d'un code (entre apostrophes, par exemple '/1') fournissant au programme un numéro correspondant au facteur auquel ce groupe de mots appartient (seulement utile pour PRINTNA = 1, et pour PUNCH = 3 ou 4). Les "mots de référence" peuvent aussi être des patterns de mots qu'on présente selon la syntaxe du programme CWPAT (et avec les contraintes de CWPAT, c'est-à-dire prévoir de mettre l'option NARR à 1). Par exemple, l'entrée suivante:

-lois-sociales: <(loi. lélgislat.) soc.> / réservé au commentaire

peut aussi être un "mot de référence" (voir le manuel pour les détails).

Syntaxe: (toujours précédée et suivie d'une apostrophe)

○ premier caractère: "/"

○ caractère suivant: un nombre de 1 à 999 (sans blanc intermédiaire). Par défaut, c'est-à-dire valeur absente ou nulle, le code est fixé à 999.

○ un commentaire (optionnel) précédé d'un ou de plusieurs blancs, par exemple
'/1 premier facteur'.

-
- Fichier de type "WORDS".

Appel de la procédure "CWSELECT":

```
... CWSELECT fn1 fn2 fn3
```

où:

- fn1 : filename du fichier de filetype "CWSELECT", "PUNCHF", "SASSEL", "SASSELOB", "PUNCHM", "SASSELSQ", et "SASSELNA"
- fn2 : filename du fichier de référence de filetype "REFER"
- fn3 : filename du fichier de filetype "WORDS" (matériel à traiter).

Le programme CWTALLY

- Fichier de type "CWTALLY" (options)
 - STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences)
issues de CSCUT ou de CRWSTRIP (associées au fichier de type
"WORDS").
 - SORTT = 'T', = défaut = total des segments
= 'S', = par segment.
 - TALLY = 'F', = défaut = tri sur la fréquence des mots,
= 'L'; = tri sur la longueur des mots.
 - SWORDS = 'chemin et nom du corpus' en entrée (fichier
"WORDS"), surtout utile pour une utilisation sur PC.
 - STEMP = 'c:...*.tmp' (défaut, STEMP = '*.tmp')

- Fichier de type "WORDS" issu de CSCUT ou de CRWSTRIP.

Appel de la procédure "CWTALLY":

```
... CWTALLY fn1 fn2
```

où:

- fn1 : filename du fichier de filetype "CWTALLY"
- fn2 : filename du fichier de filetype "WORDS".

Le programme CWWCOL

- Fichier de type "CWWCOL" (paramètres et options)

Options

- STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences)
issues de CSCUT, CRWSTRIP, ou CWEDIT (associées aux fichiers de type "WORDS")
- FMIN = 0, = défaut = fixé à 2
= n, = fréquence minimale pour qu'un mot du fichier "WORDS" de référence intervienne dans les comparaisons.
- FSEL = 'S', = défaut = FMIN s'applique à la fréquence par segment;
= 'T', = FMIN s'applique à la fréquence totale
- PROBT = 0., = défaut = fixé à 0.5
= n, = supérieur à 0 et inférieur ou égal à 1, seuil de probabilité associé au test t à partir et en-dessous duquel les résultats sont imprimés.
- SWORDS= '*' (et cette instruction renvoie à une liste de fichiers WORDS --sous la forme fn.ft.fm-- après les options mais avant les paramètres, en terminant la liste par le code * si le nombre de ces fichiers WORDS est <9).
= 'fn.ft.fm' (et cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type WORDS).
Par défaut, les fichiers doivent être détaillés dans l'appel de la procédure (cfr. infra).
- SREFER = définition du fichier "WORDS" de référence, sous la forme 'chemin et nom du fichier WORDS', surtout utile en PC.
- STEMP = 'c:...*.tmp' (défaut, STEMP = '*.tmp')

Paramètres:

○ par ligne, une définition de segment(s) en format libre, c'est-à-dire:

- [blanc(s) (~) optionnels]
- identification du fichier "WORDS" (voir *SOURCE-identification)

- [un ou plusieurs blancs (~)]
- numéro du segment

ou

en série: 2 numéros séparés par un "-" pouvant être encadré par un ou plusieurs blancs (~)

- [un ou plusieurs blancs (~)]
- l'appartenance au groupe "expérimental/contro3le", par les lettres E ou C, respectivement
- [un ou plusieurs blancs (~)]
- commentaire optionnel

- Fichier de type "WORDS" (le fichier de référence)
- Les n (de 1 à 9) fichiers de type "WORDS" contenant le matériel à traiter et correspondant aux définitions du fichier de type "CWWCOL"; un fichier par identification différente fournie.

Appel de la procédure "CWWCOL":

```
... CWWCOL fn1 fn2 fn3 [fn4...]
```

où:

- fn1 : filename du fichier de filetype "CWWCOL"
- fn2 : filename du fichier de référence de filetype "WORDS"
- fn3 : filename du premier fichier de filetype "WORDS"
(matériel à traiter)
- fn4 : filename du deuxième fichier de filetype "WORDS"
(matériel à traiter)
- etc...

Le programme CWORD

- Fichier de type "CWORD" (paramètres et options)

Options

- STAT = 0, = défaut
= 1, = impression des statistiques de tri (fréquences)
issues de CSCUT, CRWSTRIP, ou CWEDIT (associées aux fichiers de type "WORDS")

- SWORDS = '*' (et cette instruction renvoie à une liste de fichiers WORDS --sous la forme fn.ft.fm-- après les options mais avant les paramètres, en terminant la liste par le code * si le nombre de ces fichiers WORDS est <9).

- = 'fn.ft.fm' (et cette instruction donne le nom de fichier unique au cas où il n'y a qu'un seul fichier de type WORDS).

- Par défaut, les fichiers doivent être détaillés dans l'appel de la procédure (cfr. infra).

- SREFER = définition du fichier de référence, sous la forme 'chemin et nom du fichier', surtout utile en PC.

- STEMP = 'c:...*.tmp' (défaut, STEMP = '*.tmp')

Deux types de paramètres (dont le premier est nécessaire):

- par ligne, une définition de segment(s) en format libre, c'est-à-dire:

- [blanc(s) (~) optionnels]

- identification du fichier "WORDS" (voir *SOURCE-identification)

- [un ou plusieurs blancs (~)]

- numéro du segment

- ou

- en série: 2 numéros séparés par un "-" pouvant être encadré par un ou plusieurs blancs (~)

- [un ou plusieurs blancs (~)]

- l'appartenance au groupe "expérimental/contrôle", par les lettres E ou C, respectivement

- [un ou plusieurs blancs (~)]

- l'appartenance à un sous-groupe éventuel matérialisé par un chiffre de 1 à 9 (pas d'appartenance à un sous-groupe = chiffre 0)

- [un ou plusieurs blancs (~)]

- commentaire optionnel

- une définition facultative de sous-groupe en format libre (toujours par ligne), c'est-à-dire:

- [blancs (~) optionnels]

- un caractère "/"

- [blancs (~) optionnels]
 - un chiffre de 0 à 9 représentant l'échantillon total (0) ou un des 9 sous-groupes possibles
 - [un ou plusieurs blancs (~)]
 - étiquette optionnelle composée de 30 caractères
 - commentaire optionnel (après le 30ième caractère de l'étiquette)
- Fichier de type "REFER" (le fichier de référence): c'est-à-dire les différents mots de référence, entre apostrophes, et séparés par une virgule et/ou un ou plusieurs blancs (~).
 - Les n (de 1 à 9) fichiers de type "WORDS" contenant le matériel à traiter et correspondant aux définitions du fichier de type "CWORD"; un fichier par identification différente fournie.

Appel de la procédure "CWORD":

```
... CWORD fn1 fn2 fn3 [fn4...]
```

où:

- fn1 : filename du fichier de filetype "CWORD"
- fn2 : filename du fichier de référence de filetype "REFER"
- fn3 : filename du premier fichier de filetype "WORDS"
(matériel à traiter)
- fn4 : filename du deuxième fichier de filetype "WORDS"
(matériel à traiter)
- etc...

XXXVI. Documentation technique

1 Note préliminaire concernant les messages LOG et SYSLOG

Il est possible au gestionnaire de PROTAN de communiquer diverses informations via les listings produits par les programmes de PROTAN. En DOS, UNIX, ou Macintosh, cette communication se fait à l'aide d'un fichier "**PROTAN.LOG**" qui doit se trouver dans le répertoire des fichiers exécutables de PROTAN, c'est-à-dire, en DOS, parmi les fichiers de type ".EXE".

2 Le langage de programmation

Toutes les fonctions constituant le système PROTAN sont écrites en C ANSI.

3 Les fonctions externes utilisées

3.1 Brève description des fonctions

3.1.1 CPROBT : probabilité d'erreur associée au test t de Student

3.1.2 CYCVD : conversion en 'long int' d'un entier exprimé en caractères

3.1.3 CYEXIT : fin d'exécution

3.1.4 CYGETPL : options de la ligne de commande

3.1.5 CYGTAB0 : définition des jeux de caractères

3.1.6 CYPAGE : impression des deux premières lignes de l'en-tête de page

3.1.7 CYPAGES : idem pour PROTAN/STAT

3.1.8 CYSTCUT : extraction d'un item d'une ligne de texte

3.1.9 CYSTDIC : extraction d'un mot ou d'une racine pour un dictionnaire

3.1.10 CYSTREF : extraction d'un mot-clé de référence

3.1.11 CYWRCH : contrôle d'un mot ou d'une racine pour un dictionnaire

3.1.12 MEMCMPX : compilateur GNU C seulement: comparaison de chaînes de caractères non signées

3.1.13 OPTSCAN : extraction de la valeur d'une option

3.1.14 ORD2 : permutation éventuelle des octets d'un ensemble de deux

3.1.15 ORD4 : permutation éventuelle des octets d'un ensemble de quatre

3.1.16 REFSCAN : extraction d'un mot de référence

3.2 Fonctions appelées par les différents programmes

3.2.1 CDCHECK : CYCVD, CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, CYSTDIC, CYWRCH, MEMCMPX, ainsi que la fonction spécifique CDCH2

3.2.2 CDLISTA : CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN

3.2.3 CDLISTC : CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN

3.2.4 CDWJUXT : CYCVD, CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX, ainsi que la fonction spécifique CDWJ2

3.2.5 CDWLOOK : CYCVD, CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX, ainsi que les fonctions spécifiques CDWL2 et CDWL3

3.2.6 CFCHECK : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, CYSTDIC, CYWRCH, MEMCMPX

3.2.7 CFLISTA : CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN

3.2.8 CFWKWIC : CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.9 CPEXCOR : CYEXIT, CYGETPL, CYPAGES, OPTSCAN, CPROBT, MEMCMPX

3.2.10 CPFACTOR : CYEXIT, CYGETPL, CYPAGES, OPTSCAN

3.2.11 CRCHECK : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, CYSTDIC, CYWRCH, MEMCMPX

3.2.12 CRLISTA : CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.13 CRWSTRIP : CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.14 CSCHECK : CYEXIT, CYGETPL, CYGTAB0, CYSTCUT, CYPAGE, ORD2, ORD4, OPTSCAN

3.2.15 CSCUT : CYEXIT, CYGETPL, CYGTAB0, CYSTCUT, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX, ainsi que les fonctions spécifiques CSCU2 et CSCU3

3.2.16 CSEDIT : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN

3.2.17 CSJOIN : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN

3.2.18 CSSORT : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.19 CWADD : CYCVD, CYEXIT, CYGETPL, CYGTAB0, CYSTCUT, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.20 CWEDIT : CYCVD, CYEXIT, CYGETPL, CYGTAB0, CYSTCUT, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.21 CWFLOW : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.22 CWKWIC : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, REFSCAN, CYSTREF, MEMCMPX

3.2.23 CWKWOC : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, REFSCAN, CYSTREF, MEMCMPX

3.2.24 CWNEW : CYCVD, CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.25 CWPAT : CYEXIT, CYGETPL, CYGTAB0, CYPAGE, ORD2, ORD4, OPTSCAN, CYSTDIC, MEMCMPX

3.2.26 CWREFER : CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN

3.2.27 CWSELECT : CYCVD, CYEXIT, CYGETPL, CYGTAB0, CYSTCUT, CYPAGE, ORD2, ORD4, OPTSCAN, REFSCAN, CYSTDIC, MEMCMPX, ainsi que les fonctions spécifiques CWSE2 et CWSE3

3.2.28 CWTALLY : CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN, MEMCMPX

3.2.29 CWWCOL : CYCVD, CYEXIT, CYGETPL, CYPAGE, ORD2, ORD4, OPTSCAN, CPROBT, MEMCMPX

3.2.30 CWORD : CYCVD, CYEXIT, CYGETPL, CYGTAB0, CYSTCUT, CYPAGE, ORD2, ORD4, OPTSCAN, REFSCAN, CPROBT, MEMCMPX

3.3 Remarques

Les fichiers "système" WORDS, DICTS, ROOTS et FINDS ainsi que les fonctions de tri ont une représentation arithmétique basée sur une structure hexadécimale à 1, 2 ou 3 octets suivant l'amplitude possible des nombres (cf. aussi les fonctions ORD2 et ORD4).

3.4 Fichiers généraux utilisés par PROTAN

Tous les programmes utilisent SYSPRIN (sortie, texte, 80 car.) pour la liste à imprimer, SYSLOG (entrée, texte, 80 car.) pour l'affichage éventuel sur SYSPRIN d'informations générales sur le système, et MASTER (entrée, texte, 80 car.) pour l'introduction des options.

3.4.1 Fichiers utilisés par CDCHECK

3.4.1.1 Entrée

3.4.1.1.1 INDICT (chaîne de fichiers, texte, 80 car.)

3.4.1.2 Sortie

3.4.1.2.1 DICTS (binaire, 40 car.)

3.4.1.3 Travail

3.4.1.3.1 SYSUT4 (binaire, 40 car.)

3.4.2 Fichiers utilisés par **CDLISTA**

3.4.2.1 Entrée

3.4.2.1.1 DICTS (binaire, 40 car.)

3.4.3 Fichiers utilisés par **CDLISTC**

3.4.3.1 Entrée

3.4.3.1.1 DICTS (binaire, 40 car.)

3.4.4 Fichiers utilisés par **CDWJXT**

3.4.4.1 Entrée

3.4.4.1.1 DICTS1 (binaire, 40 car.)

3.4.4.1.2 DICTS2 (optionnel, binaire, 40 car.)

3.4.4.1.3 WORDS (binaire, 40 car.)

3.4.4.2 Sortie

3.4.4.2.1 SYSPUN (texte, 80 car., puis longueur non ajustée)

3.4.4.3 Travail

3.4.4.3.1 SYSUT2 (texte, 80 car.)

3.4.4.3.2 SYSUT3 (binaire, 25 car.)

3.4.4.3.3 SYSUT3P (binaire, 29 car.)

3.4.4.3.4 SYSUT4 (binaire, 12 car.)

3.4.5 Fichiers utilisés par **CDWLOOK**

3.4.5.1 Entrée

3.4.5.1.1 DICTS (binaire, 40 car.)

3.4.5.1.2 WORDS (binaire, 40 car.)

3.4.5.2 Sortie

3.4.5.2.1 SYSPUN (texte, 80 car., puis longueur non ajustée)

3.4.5.3 Travail

3.4.5.3.1 SYSOUT (texte, 80 car.)

3.4.5.3.2 SYSUT3 (binaire, 25 car.)

3.4.5.3.3 SYSUT3P (binaire, 29 car.)

3.4.5.3.4 SYSUT4 (binaire, 10 car.)

3.4.6 Fichiers utilisés par **CFCHECK**

3.4.6.1 Entrée

3.4.6.1.1 INFIND (chaîne de fichiers,
texte, 80 car.)

3.4.6.2 Sortie

3.4.6.2.1 FINDS (binaire, 50 car.)

3.4.6.3 Travail

3.4.6.3.1 SYSUT4 (binaire, 50 car.)

3.4.7 Fichiers utilisés par **CFLISTA**

3.4.7.1 Entrée

3.4.7.1.1 FINDS (binaire, 50 car.)

3.4.8 Fichiers utilisés par **CFWKWIC**

3.4.8.1 Entrée

3.4.8.1.1 FINDS (binaire, 50 car.)

3.4.8.1.2 WORDS (binaire, 40 car.)

3.4.8.2 Sortie

3.4.8.2.1 SYSPUN (texte, 80 car.)

3.4.8.3 Travail

3.4.8.3.1 SYSUT2 (binaire, 27 car.)

3.4.8.3.2 SYSUT3 (binaire, 27 car.)

3.4.8.3.3 SYSUT4 (binaire, 65 car.)

3.4.9 Fichiers utilisés par **CPEXCOR**

3.4.9.1 Entrée

3.4.9.1.1 PUNCHF (texte, 80 car., puis
longueur non ajustée)

3.4.9.2 Sortie

3.4.9.2.1 SYSPUN (texte, 80 car.)

3.4.10 Fichiers utilisés par **CPFACTOR**

3.4.10.1 Entrée

3.4.10.1.1 PUNCHF (texte, 80 car., puis longueur non ajustée)

3.4.10.2 Travail

3.4.10.2.1 SYSUT4 (binaire, 4 car.)

3.4.11 Fichiers utilisés par **CRCHECK**

3.4.11.1 Entrée

3.4.11.1.1 INROOT (chaîne de fichiers, texte, 80 car.)

3.4.11.2 Sortie

3.4.11.2.1 ROOTS (binaire, 50 car.)

3.4.11.3 Travail

3.4.11.3.1 SYSUT4 (binaire, 50 car.)

3.4.11.3.2 SYSUT5 (binaire, 50 car.)

3.4.12 Fichiers utilisés par **CRLISTA**

3.4.12.1 Entrée

3.4.12.1.1 ROOTS (binaire, 50 car.)

3.4.13 Fichiers utilisés par **CRWSTRIP**

3.4.13.1 Entrée

3.4.13.1.1 ROOTS (binaire, 50 car.)

3.4.13.1.2 WORDS (binaire, 40 car.)

3.4.13.2 Sortie

3.4.13.2.1 WORDSO (binaire, 40 car.)

3.4.13.2.2 SYSPUN (texte, 40 car., puis longueur non ajustée)

3.4.13.3 Travail

3.4.13.3.1 SYSUT2 (binaire, 6 car.)

3.4.13.3.2 SYSUT4 (binaire, 35 car.)

3.4.14 Fichiers utilisés par **CSCHECK**

3.4.14.1 Entrée

3.4.14.1.1 SOURCE (chaîne de fichiers,
texte, 80 car.)

3.4.15 Fichiers utilisés par **CSCUT**

3.4.15.1 Entrée

3.4.15.1.1 SOURCE (chaîne de fichiers,
texte, 80 car.)

3.4.15.2 Sortie

3.4.15.2.1 WORDS (binaire, 40 car.)

3.4.15.2.2 SYSPUN (texte, 40 car., puis
longueur non ajustée)

3.4.15.2.3 PUNSTAT (texte, 80 car., puis
longueur non ajustée)

3.4.15.3 Travail

3.4.15.3.1 SYSUT2 (binaire, 80 car.)

3.4.15.3.2 SYSUT3 (binaire, 4 car.)

3.4.15.3.3 SYSUT4 (binaire, 28 car.)

3.4.16 Fichiers utilisés par **CSEDIT**

3.4.16.1 Entrée

3.4.16.1.1 SOURCE (chaîne de fichiers,
texte, 80 car.)

3.4.16.2 Sortie

3.4.16.2.1 SOURCEO (texte, 80 car.)

3.4.17 Fichiers utilisés par **CSJOIN**

3.4.17.1 Entrée

3.4.17.1.1 SOURCE (chaîne de fichiers,
texte, 80 car.)

3.4.17.2 Sortie

3.4.17.2.1 SOURCEO (texte, 80 car.)

3.4.18 Fichiers utilisés par **CSSORT**

3.4.18.1 Entrée

3.4.18.1.1 SOURCE (chaîne de fichiers,
texte, 80 car.)

3.4.18.2 Sortie

3.4.18.2.1 SOURCEO (texte, 80 car.)

3.4.18.3 Travail

3.4.18.3.1 SYSUT4 (binaire, 95 car.)

3.4.19 Fichiers utilisés par **CWADD**

3.4.19.1 Entrée

3.4.19.1.1 WORDS (binaire, 40 car.)

3.4.19.2 Sortie

3.4.19.2.1 WORDSO (binaire, 40 car.)

3.4.19.3 Travail

3.4.19.3.1 PARMOUT (binaire, 27 car.)

3.4.20 Fichiers utilisés par **CWEDIT**

3.4.20.1 Entrée

3.4.20.1.1 WORDS (binaire, 40 car.)

3.4.20.2 Sortie

3.4.20.2.1 WORDSO (binaire, 40 car.)

3.4.20.2.2 SYSPUN (texte, 40 car., puis
longueur non ajustée)

3.4.20.3 Travail

3.4.20.3.1 SYSUT2 (binaire, 6 car.)

3.4.20.3.2 SYSUT4 (binaire, 35 car.)

3.4.20.3.3 PARMOUT (binaire, 67 car.)

3.4.20.3.4 WRDSOUT (binaire, 40 car.)

3.4.21 Fichiers utilisés par **CWFLOW**

3.4.21.1 Entrée

3.4.21.1.1 WORDS (binaire, 40 car.)

3.4.21.2 Sortie

3.4.21.2.1 SYSPUN (texte, 80 car., puis
longueur non ajustée)

3.4.21.3 Travail

3.4.21.3.1 SYSUT3 (binaire, 29 car.)

3.4.22 Fichiers utilisés par **CWKWIC**

3.4.22.1 Entrée

3.4.22.1.1 WORDS (binaire, 40 car.)

3.4.22.2 Travail

3.4.22.2.1 SYSUT3 (binaire, 90 car.)

3.4.22.2.2 SYSUT4 (binaire, 28 car.)

3.4.23 Fichiers utilisés par **CWKWOC**

3.4.23.1 Entrée

3.4.23.1.1 WORDS (binaire, 40 car.)

3.4.23.2 Sortie

3.4.23.2.1 SYSPUN (binaire, 40 car., puis
longueur non ajustée)

3.4.23.2.2 REFER (texte, 80 car.)

3.4.23.2.3 INFIND (texte, 80 car.)

3.4.23.2.4 INROOT (texte, 80 car.)

3.4.23.3 Travail

3.4.23.3.1 SYSUT4 (binaire, 32 car.)

3.4.24 Fichiers utilisés par **CWNEW**

3.4.24.1 Entrée

3.4.24.1.1 WORDS (binaire, 40 car.)

3.4.24.2 Sortie

3.4.24.2.1 SYSPUN (texte, 80 car., puis
longueur non ajustée)

3.4.24.3 Travail

3.4.24.3.1 SYSUT2 (binaire, 11 car.)

3.4.24.3.2 SYSUT3 (binaire, 20 car.)

3.4.24.3.3 SYSUT4 (binaire, 27 car.)

3.4.25 Fichiers utilisés par **CWPAT**

3.4.25.1 Entrée

3.4.25.1.1 WORDS (binaire, 40 car.)

3.4.25.2 Travail

3.4.25.2.1 SYSUT3 (binaire, 30 car.)

3.4.25.2.2 SYSUT4 (binaire, 29 car.)

3.4.25.2.3 SYSUT5 (binaire, 29 car.)

3.4.26 Fichiers utilisés par **CWREFER**

3.4.26.1 Entrée

3.4.26.1.1 WORDS (binaire, 40 car.)

3.4.26.2 Sortie

3.4.26.2.1 SYSPUN (texte, 80 car.)

3.4.27 Fichiers utilisés par **CWSELECT**

3.4.27.1 Entrée

3.4.27.1.1 REFER (texte, 80 car.)

3.4.27.1.2 WORDS (binaire, 40 car.)

3.4.27.2 Sortie

3.4.27.2.1 SYSPUN (texte, 80 car., puis
longueur non ajustée)

3.4.27.3 Travail

3.4.27.3.1 NVER (binaire, 40 car.)

3.4.27.3.2 PARMOUT (texte, 80 car.)

3.4.27.3.3 SYSUT3 (binaire, 25 car.)

3.4.27.3.4 SYSUT3P (binaire, 29 car.)

3.4.27.3.5 SYSUT4 (binaire, 10 car.)

3.4.27.3.6 SYSUT5 (binaire, 29 car.)

3.4.27.3.7 SYSUT6 (binaire, 30 car.)

3.4.28 Fichiers utilisés par **CWTALLY**

3.4.28.1 Entrée

3.4.28.1.1 WORDS (binaire, 40 car.)

3.4.28.2 Travail

3.4.28.2.1 SYSUT4 (binaire, 27 car.)

3.4.29 Fichiers utilisés par **CWWCOL**

3.4.29.1 Entrée

3.4.29.1.1 REFER (binaire, 40 car.)

3.4.29.1.2 WORDS (chaîne de fichiers,
binaire, 40 car.)

3.4.29.2 Travail

3.4.29.2.1 SYSUT2 (binaire, 14 car.)

3.4.29.2.2 SYSUT5 (binaire, 26 car.)

3.4.29.2.3 REFROUT (binaire, 20 car.)

3.4.30 Fichiers utilisés par **CWORD**

3.4.30.1 Entrée

3.4.30.1.1 REFER (texte, 80 car.)

3.4.30.1.2 WORDS (chaîne de fichiers,
binaire, 40 car.)

3.4.30.2 Travail

3.4.30.2.1 SYSUT2 (binaire, 14 car.)

3.4.30.2.2 SYSUT5 (binaire, 26 car.)

3.4.30.2.3 REFROUT (binaire, 20 car.)

XXXVII. Références

- Allport, G. W. The use of personal documents in psychological science. New York: Social Science Research Council, 1942.
- Averill, J. R. A semantic atlas of emotional concepts. JSAS Catalogue of Selected Documents in Psychology, 1975, 5, 330. (Ms. No. 421).
- Bakeman, R., & Gottman, J. M. Observing interaction: An introduction to sequential analysis. Cambridge, Mass.: Cambridge University Press, 1986.
- Bucci, W. S. (1985). Dual coding: A cognitive model for psychoanalytic research. Journal of the American Psychoanalytic Association, 33, 571-607.
- Daoust, F. SATO (Système d'analyse de texte par ordinateur), Version 3.6. Manuel de référence. Université du Québec à Montréal: Centre d'Analyse de textes par ordinateur (ATO), 1992.
- Deese, J. American freedom and the social sciences. New York: Columbia University Press, 1985.
- Delcroix, C. Comparaison de contenus de rêves masculins et féminins. Louvain-la-Neuve, Université Catholique de Louvain, Faculté de Psychologie et des Sciences de l'Education, Séminaire d'Analyse du Contenu 1985-1986, non-publié, 1986, 8 pages.
- Descamps, A. Essai de construction d'un dictionnaire d'imagerie suicidaire. Louvain-la-Neuve, Université Catholique de Louvain, Faculté de Psychologie et des Sciences de l'Education, Mémoire de Licence non-publié, 1982.
- Desrochers, A., & Bergeron, M. (1992). Valeurs de fréquence subjective et d'imagerie pour un échantillon de 1,916 substantifs de la langue française (Manuscript No. RR36). Laboratoire de Psychologie Cognitive, Ecole de Psychologie, Université D'Ottawa, Ottawa, Ontario K1N 6N5, Canada (Damch@acadvm1.uottawa.ca).
- Ericsson, K. A., & Simon, H. A. Protocol analysis: Verbal reports as data. Cambridge, MA.: The M.I.T. Press, 1984.
- Foucault, M. L'archéologie du savoir. Paris: Gallimard, 1969.
- Fuller, W. A. Introduction to statistical time series. New York: Wiley, 1976.
- Guillaumont, A., & Minel, J.-L., MEDIUM: Realities and projects. Computers and the Humanities, 1986, 20, 269-271.
- Gunning, R. The technic of clear writing. New York: MacGraw Hill, 1952.
- Guntern, A. Glauben-Typen und ihre Verankerung im Menschsein. Eine Inhaltanalyse der religiösen Rede mit Hilfe eines automatisierten empirischen Verfahrens. Louvain-la-Neuve, Université Catholique de Louvain, Faculté de Psychologie et des Scien-

- ces de l'Education, Thèse de Doctorat non-publiée, 1981, 306 pages.
- Hogenraad, R. Analyse du contenu et contenus mentaux. Louvain-la-Neuve:CIACO, 1985.
- Hogenraad, R. A little organon of content analysis. Louvain-la-Neuve, Faculté de Psychologie et des Sciences de l'Education, Université Catholique de Louvain, Thèse d'Agrégation de l'Enseignement Supérieur, 1990.
- Hogenraad, R., & Bestgen, Y. On the thread of discourse: Homogeneity, trends, and rhythms in texts. Empirical Studies of the Arts, 1989, 7:1, 1-22.
- Hogenraad, R., Bestgen, Y., & Nysten, J. L. Terrorist rhetoric: Texture and architecture. In E. Nissan & K. M. Schmidt (Eds.) From information to knowledge. Conceptual and content analysis by computer (pp. 54-67). Oxford, England: Intellect Books, 1995.
- Hogenraad, R., & Orianne, E. Valences d'imagerie de 1.130 noms de la langue française parlée. Psychologica Belgica, 1981, 21, 21-30.
- Hogenraad, R., & Orianne, E. "A penny for your thoughts": Imagery value and periodicity of interior monologue. Journal of Mental Imagery, 1983, 7:1, 51-62.
- Hogenraad, R., & Orianne, E. Imagery, regressive thinking, and verbal performance in internal monologue. Imagination, Cognition, and Personality, 1985-1986, 5:2, 127-145.
- Hostier, J. P. Un atlas sémantique de concepts d'agressivité. Normes et validation. Louvain-la-Neuve, Université Catholique de Louvain, Faculté de Psychologie et des Sciences de l'Education, Mémoire de Licence non-publié, 1987.
- Iker, H. P., & Klein, R. H. WORDS: A computer system for the analysis of content. Behavior Research Methods and Instrumentation, 1974, 6, 430-438.
- Iker, H. P. SELECT: A computer program to identify associationally rich words for content analysis. I. Statistical results. Computers and the Humanities, 1974, 8, 313-319.
- Iker, H. P. SELECT: A computer program to identify associationally rich words for content analysis. I. Substantive results. Computers and the Humanities, 1975, 9, 3-12.
- Kant, E. Critique de la raison pure.
- Krippendorff, K. Content analysis: An introduction to its methodology. Beverly Hills: Sage, 1980.
- Leleu, S. Un atlas sémantique de concepts d'émotion. Normes et validation. Louvain-la-Neuve, Université Catholique de Louvain, Faculté de Psychologie et des Sciences de l'Education, Mémoire de Licence non-publié, 1987.

-
- Mailloux, S. L., Johnson, M. E., Fisher, D. G., & Pettibone, T. J. How reliable is computerized assessment of readability? Computers in Nursing, 1995, 13:5, 221-225.
- Martindale, C. Romantic progression: The psychology of literary history. Washington, D.C.: Hemisphere, 1975.
- Martindale, C. The night journey: Trends in the content of narratives symbolizing alterations of consciousness. Journal of Altered States of Consciousness, 1979, 4, 321-343.
- McCann, S. J. H., & Stewin, L. L. (1990). Good and bad years: An index of American social, economic, and political threat (1920-1986). Journal of Psychology, 124, 601-617.
- Mounin, G. Dictionnaire de la linguistique. Paris: Presses Universitaires de France, 1974.
- Muskens, G. Mathematical analysis of content. Quality and Quantity, 1985, 19, 99-103.
- Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. The measurement of meaning. Urbana, Ill.: The University of Illinois Press, 1957.
- Paivio, A., Yuille, J. C., & Madigan, S. A. Concreteness, imagery, and meaningfulness values for 925 nouns. Journal of Experimental Psychology Monograph Supplement, 1968, 76, 1-125.
- Reimer, Stephen R. LitStats, V1.62. Edmonton, Canada: Computer Software, 1992.
- Richeaudeau, F. La lisibilité. Paris: Editions Retz, 1969.
- Sarbin, T. R. (ed.) Narrative psychology: The storied nature of human conduct. New York: Preager, 1986.
- Saris-Gallhofer, I. N., Saris, W. E., & Morton, E. L. A validation study of Holsti's content analysis procedure. Quality and Quantity, 1978, 12, 131-145.
- Spence, D. P. PL/I programs for content analysis. Behavioral Science, 1969, 14, 432-433.
- Spence, D. P. Narrative truth and historical truth. Meaning and interpretation in psychoanalysis. New York: W. W. Norton, 1982.
- Spence, D. P. GENERAL ANALYZER: User's Manual. Community Mental Health Center, Rutgers Medical School, Piscataway, N.J. Unpublished manuscript, no date, 43 pages.
- Spence, D. P., Scarborough, H. S., & Ginsberg, E. H. Lexical correlates of cervical cancer. Social Science and Medicine, 1978, 12, 141-145.
- Spence, D. P. Lawfulness in lexical choice: A natural experiment. Journal of the American Psychoanalytical Association, 1979.

-
- Stone, P. J. Standards for computer-aided content analysis: The Pisa convention and recommendations. Informations sur les Sciences Sociales, 1974, 14, 127-137:
- Stone, P. J., Dunphy, D. C., Smith, M. S., & Ogilvie, D. M. The GENERAL INQUIRER: A computer approach to content analysis. Cambridge, MA.: The M.I.T. Press, 1966.
- Sweeney, K. & Whissell, C. A dictionary of affect in language: I. Establishment and preliminary validation. Perceptual and Motor Skills, 1984, 59, 695-698.
- Valéry, P. "Mon Faust" (Lust) (1945). Paris: Gallimard, 1946.
- van Eeckhoudt, C. La régression comme concept et instrument au service de l'analyse du contenu. Louvain-la-Neuve, Université Catholique de Louvain, Faculté de Psychologie et des Sciences de l'Education, Mémoire de Licence non-publié, 1981, 114 pages.
- Vygotsky, L. Thought and language. Cambridge, MA.: The M.I.T. Press, 1962. (Ouvrage original publié en 1934).
- Weber, R. P. Measurement models for content analysis. Quality and Quantity, 1983, 17, 127-149.
- Weber, R. P. Basic content analysis. Beverly Hills, CA.: Sage Publications, 1985.
- Weber, R. P. Correlational models of content. Reply. Quality and Quantity, 1986, 20, 273-276.
- Whissell, C. & Charuk, K. A dictionary of affect in language: II. Word inclusion and additional validation. Perceptual and Motor Skills, 1985, 61, 65-66.
- Whissell, C., Fournier, M., Pellard, R., Weir, D., & Makarec, K. A dictionary of affect in language: IV. Reliability, validity, and applications. Perceptual and Motor Skill, 1986, 62, 875-881.